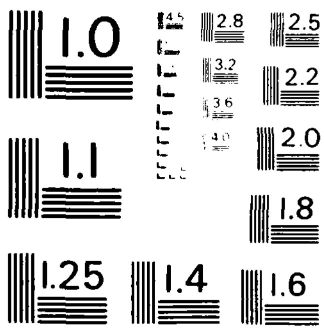
AD-A161 205 A VALIDATION OF AN ACCOUNTING UPPER ERROR LIMIT BOUND 1/2
(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH
SCHOOL OF SYSTEMS AND LOGISTICS M W HELTON SEP 85
UNCLASSIFIED AFIT/GSM/LSY/85S-17 F/G 5/1 NL


1.0
2.8
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6

AD-A161 205

DTIC FILE COPY

A VALIDATION OF AN ACCOUNTING UPPER

ERROR LIMIT BOUND

THESIS

Michael W. Helton
First Lieutenant, USAF

AFIT/GSM/LSY/85S-17

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

85 11 12 105

2

AFIT/GSM/LSY/85

A VALIDATION OF AN ACCOUNTING UPPER

ERROR LIMIT BOUND

THESIS

Michael W. Helton
First Lieutenant, USAF

AFIT/GSM/LSY/85S-17

The contents of the document are technically accurate, and no sensitive items, detrimental ideas, or deleterious information are contained therein. Furthermore, the views expressed in the document are those of the author(s) and do not necessarily reflect the views of the School of Systems and Logistics, the Air University, the United States Air Force, or the Department of Defense.

AFIT/GSM/LSY/85S-17

A VALIDATION OF AN ACCOUNTING UPPER ERROR LIMIT BOUND

THESIS

Presented to the Faculty of the School of Systems and Logistics

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Systems Management

Michael W. Helton, B.S.

First Lieutenant, USAF

September 1985

Approved for public release; distribution unlimited

Acknowledgments

I wish to thank Jeffrey J. Phillips, my thesis advisor, for his helpful comments, encouragement, and assistance throughout this research effort. I also wish to give special thanks to my wife, Chris, for her understanding and patience throughout the accomplishment of my thesis.

Michael W. Helton

Table of Contents

Page

List of Figures

List of Tables

Abstract

The purpose of this research was to examine the validity of an accounting upper error limit bound. The bound examined was the DUS-cell method suggested by Leslie, Teitlebaum, and Anderson which was supposed to reduce bound conservativism and produce actual confidence levels closer to the nominal confidence levels.

The analysis was accomplished by examining the robustness, the relative tightness, and the effects of error rate, error clustering, mean taint, and error amount intensity on the coverage and relative tightness of the DUS-cell bound. The results of this research indicate that the DUS-cell bound is not robust and is tighter than the Stringer bound. The results also demonstrated that error rate has the greatest effect on coverage and relative tightness, error clustering has some effect, and error taint has little effect. The results also indicate that error amount intensity, a population characteristic, affects coverage and relative tightness of the DUS-cell bound significantly.

A VALIDATION OF AN ACCOUNTING UPPER ERROR LIMIT BOUND

I. INTRODUCTION

General Problem

Because the Air Force has become so large, auditors can no longer audit every account within accounting populations such as accounts receivable or inventory accounts. Therefore, they must rely on some type of method that allows them to estimate, as accurately as possible, the maximum amount by which an account is in error or the upper error limit. The upper error limit that is computed using a prescribed method is then compared to a value that constitutes a material error, thus enabling the auditor to accept or reject the book value as being a reasonable representation of the correct amount (1:180).

One of the major problems in predicting upper error limits is that the characteristics of accounting populations are virtually unknown. Assuming that "the number of items in error in a population of accounts is likely to be very small, the auditor, even with a large sample" may gain little information about the true population error (1:180). Because of the inability to directly extract information about the accounting population, the upper error limit bound

may be much larger than the true error. When the upper error limit is larger than a material error, the auditor must reject the hypothesis that the book value of the population is a reasonable representation of the true value. Therefore, the auditor would require his client to adjust the book value of the population to reflect the material error. Even though few published reports exist on the actual characteristics of accounting populations, it is generally assumed that accounting populations contain relatively few errors which prohibits the use of traditional error estimating methods (2:270).

A second major problem involves the use of classical sampling techniques on accounting populations.

> When auditors first considered the use of statistical sampling techniques for estimating the total audit amount for a population (or, equivalently, the total amount of error for a population), they turned to the classical sample survey techniques [3:77].

However, classical sampling techniques led to several difficulties with trying to infer, from the sample, the characteristics of the population. Subsequently,

> Stringer [1963] cautioned that the use of these estimators entails potential dangers because their estimated standard errors will equal zero when no errors are found in the sample. With classical large-sample confidence limits this would imply that the point estimate of the total audit amount or the total error amount is perfect, which, obviously, is not a reasonable conclusion in an audit situation employing a moderately large sample size [3:77].

In addition to Stringers caution, Kaplan demonstrated that another difficulty with classical sampling techniques

was that "the nominal confidence level implied by the normal distribution for large-sample confidence intervals was frequently far different from the actual proportion of correct confidence intervals..." (3:77). He also pointed out that classical sampling techniques using ratio, difference and regression estimators are methods "designed for a homogeneous population, rather than for a typical accounting population containing a large number of correct items and a small number of misstated items" (4:670). Neter and Loebbecke went a step further than Kaplan and concluded that

> the nominal confidence levels implied by large-sample normal theory..., for both the unstratified and stratified sampling, often differed greatly from the actual proportions of correct confidence intervals when the population error rate was small to moderate [3:78].

Other problems associated with classical sampling techniques are that the methods ignore "process variability and takes into account only the variation due to the sampling design... [and] classical statistical methods do not explicitly incorporate prior subjective information" (5:306-307).

Because many auditors believe that they should be allowed to interject subjective information in the process, they have turned to Bayesian or subjective sampling methods. These Bayesian methods, as opposed to the classical models, require the auditor to use prior knowledge in obtaining the upper error limit. This desire itself has led to a third

major problem. Godfrey and Andrews have demonstrated that there are two sources of variability. One source is due to the process and the other is due to sampling. Process variability results from the fact that the populations are not homogeneous and the sampling variability is due to the fact that auditors are only sampling a subset of the population (5:306). In essence, the problem is not that the models do not let auditors incorporate their prior knowledge, but instead, the problem is in assessing the distribution of this prior knowledge (5:313).

Specific Problem

Although there have been several attempts at developing methods to deal with unknown accounting population characteristics, the inability of classical sampling techniques to deal with those populations, and unknown distributions of prior knowledge about these populations, no one method has completely succeeded. Most of the methods have overcome some part of each problem to a partial degree, but none have been able to produce results which capture the dollar error amount in the accounting population at a level of confidence that is close to the nominal confidence level. The methods that are currently in existence produce actual confidence levels far greater than the nominal levels of confidence. In other words, the methods employed today produce bounds that are larger than they need to be which, in turn, causes the auditor to audit more accounts than

necessary. This concept is known as bound conservativism. However, Leslie, Teitlebaum, and Anderson suggested a method known as dollar unit sample (DUS)-cell bound which was supposed to reduce bound conservatism and produce actual confidence levels that are closer to nominal levels, allowing auditors to audit fewer accounts.

Definition of Key Terms

Conservativism - that is, at the chosen confidence level, the upper error limit will be slightly overstated or, conversely, for the computed upper error limit, the actual confidence level achieved will be higher than the nominal level of the bound (6:129).

Nominal Confidence Level - the expected frequency of correct intervals or bounds in repeated samples (7:6).

Actual Confidence Level - the actual frequency of correct intervals or bounds in repeated samples. Often referred to as the coverage of the bound.

Error Rate - the probability that an individual population item is in error.

Coverage - is the percent of correct upper error limit bounds out of the number sampled. To be correct the calculated upper error limit bound must be equal to or greater than the true dollar error in the population.

Error Taint - the proportionate error, that is, the error amount divided by the book value of the line item (2:281).

Mean Taint - the average taint of the population.

Robustness - a bound is robust if it provides, for each and every population, an actual coverage that is equal to or greater than the nominal level of confidence.

Modified Bound - projects the error in the sample to the unsampled dollars and then adds to that projection the amount of error present in the sample.

Error Clustering - the grouping of errors by line items according to book values of accounts in a population.

Relative Tightness - is the tightness of an upper error limit bound relative to a more conservative upper error limit bound.

Error Amount Intensity - is the ratio of the total population dollar error to the total population book value.

Research Questions

The following research questions were developed to accomplish the analysis of the DUS-cell evaluation method suggested by Leslie, Teitlebaum, and Anderson.

1. Are the Stringer, modified Stringer, DUS-cell, and modified DUS-cell bounds robust? The objective is to examine the bounds to see if they provide a high degree of coverage. Put another way, do the bounds have an actual level of confidence that is higher than the nominal confidence level?
2. Does the DUS-cell bound provide the auditor with a tighter bound than the Stringer bound? The purpose

for asking this question is to determine if the DUS-cell method yields a tighter confidence interval than the Stringer bound which would reduce the potential of rejecting populations that are not materially in error.

3. Does the level of confidence affect the robustness of the modified or unmodified DUS-cell bound? The reason for examining the effect of different confidence levels is that the bound may not perform the same under different levels of confidence corresponding to assumptions of greater risk. If so , it would be important for the auditor to be aware of this fact.

4. Which characteristics of accounting populations have a significant impact on the performance of the DUS-cell method in calculating the upper error limit. In other words, do error rate, error clustering, error taint, or error amount intensity have an impact on the performance of the DUS-cell method? The objective is to determine if the distribution, size, or number of errors in an accounting population affect the DUS-cell bound.

Literature Review

The literature review will present, in chronological order, those major studies that have examined upper error limit methods. Also, it will review the development of

dollar unit sampling and studies that have attempted to describe the characteristics of errors in accounting populations. Finally, it will highlight both Bayesian and non-Bayesian upper error limit bounds that have been suggested to overcome the problem of conservative bounds.

Stringer was one of the earliest individuals to caution that the use of classical sampling techniques was potentially dangerous because "their estimated standard errors will equal zero when no errors are found in the sample" (3:77). Consequently, he suggested his own method for predicting the upper error limit. Although the Stringer bound does work it has two major drawbacks. First and most significantly, there is "no general statistical theory supporting the confidence level attributed to the procedure" (3:78). Secondly, the confidence interval attained is far too conservative. In particular, it produces upper error limits far above the true error of the population. Although the Stringer bound contains these two flaws it has been and continues to be the baseline for comparison with other methods, both Bayesian and non-Bayesian.

During this same time frame, a sampling technique was being developed to compensate for the suspected low error rates in accounting populations. This sampling procedure is Dollar Unit Sampling (DUS) sometimes referred to as Monetary Unit Sampling (MUS). In this paper it will be referred to as DUS and it is well described by Garstka who stated that

DUS samples individual dollars in the population, rather than entire items. On the basis of the information contained in the sample, inferences are drawn about the maximum percentage of dollars that are in error in the population, and an upper error limit is calculated [1:181].

Although the statistical support for this method is suspect, it has become widely accepted and used because the results are broadly correct (8:126). Even though DUS gave auditors a new sampling technique to better represent the population as a whole the confidence levels were still far too conservative. In order to further reduce the actual confidence so that it was closer to the nominal confidence level DUS was performed in conjunction with methods of evaluating the dollar unit sampled. "Anderson and Teitlebaum [1973] proposed the use of dollar unit sampling with the Stringer bound" (3:78). However, the Stringer bound was still found to be highly conservative in simulation studies conducted by Reneau and Leitch et al (7:2).

Another bound for evaluating sampled dollar units was proposed by Neter, Leitch, and Fienburg in 1978. They named their bound the multinomial bound because it uses a multinomial distribution. They claim that their approach is "nonparametric in nature and... it has known characteristics so that the auditor is assured of the specified confidence level regardless of the nature of the population and the nature of the error pattern" (3:77).

Evaluation of the multinomial bound has demonstrated that it produces upper error limits "significantly tighter than the Stringer bound, with actual confidence levels nearer the nominal level" (7C:3). Furthermore, in a later study by Godfrey and Neter it was pointed out that

> While the use of monetary-unit-sampling with either the Stringer bound or the multinomial bound avoids the problem of actual confidence levels far below the nominal level when there are relatively few errors in an accounting population, neither the Stringer bound nor the multinomial bound permit a direct netting of over- and understatement errors. Further, neither of these two bounds directly takes into account prior information which the auditor may have about the population error rate... [7:3].

In 1979 D.R. Cox and E.J. Snell offered a Bayesian approach to calculate the upper error limit. A Bayesian approach is when the auditor is allowed to interject subjective probabilities based on a personal assessment of the likely occurrence of a particular event (9:791). The Cox and Snell method uses infinite population sample theory as a theoretical foundation for the development of a Bayesian upper bound for the total overstatement error in the population. Their bound includes two prior probability parameters. The first is error rate, which they assume follows a gamma prior distribution. The second parameter is mean taint, where the reciprocal is assumed to follow a gamma prior distribution. They also make the assumption that these two parameters(error rate and mean taint) are independent variables (8:125-132;7:8-9). However, they did not attempt to validate their approach in their paper. One

likely reason for failing to validate their method is the lack of information on the characteristics of errors in accounting populations.

In a study conducted by Johnson, Leitch, and Neter in 1981, they analyzed the error rates, examined the distributions of the error amounts, and studied the error taints in accounting populations. Their findings, based on 55 accounts receivables and 26 inventory populations were:

> 1. There is great variability in the error rates for both types of audits [populations], with the error rates in the inventory audits [populations] tending to be substantially higher than those for accounts receivable.
> 2. There is some evidence from the 81 audits [populations] which suggests that the error rates in both types of audits [populations] may be higher for the larger accounts and for accounts with larger line items than for other accounts....
> 3. An examination of within-audit [population] data for 20 audits [populations] provides strong evidence that the error rate tends to increase for the larger line items for both receivables and inventory audits [populations].
> 4. Most errors in receivables audits [populations] are overstatement errors, while in inventory audits overstatements and understatements are more balanced in numbers.
> 5. An analysis of the 20 audits [populations] with the largest numbers of errors indicated that receivables errors tend to be larger and less variable than inventory errors. The distributions of error amounts for both types of audits [populations] are far from normal, exhibiting both greater peakedness near the mean and "fatter" tails in the upper direction. Most of the distributions are positively skewed, and the standardized distributions of the error amounts for each type of audit tend to be highly similar.
> 6. The distributions of error taintings are characterized by pronounced discontinuities at +100 percent, particularly so for receivables audits [populations] where 100 percent overstatement errors are frequently present to a large extent.
> 7. The tainting distributions for inventories show large negative taintings which occur with

> moderate frequency, the maximum often beyond -300 percent. The mean taintings for inventories are smaller than those for receivables audits [populations]. The mean taintings for receivables are surprisingly large, in half the audits [populations] exceeding 48 percent.
>
> 8. The distributions of taintings are variable for both types of audits [populations], especially those for inventory audits [populations], and depart substantially from a normal distribution. Quite a few of the tainting distributions for inventories are negatively skewed [2:291].

In light of the problems with the Stringer bound and the multinomial bound along with the empirical findings presented in the Neter, Leitch, and Johnson study, Godfrey and Andrews introduced "an alternative procedure for establishing an upper precision limit (UPL) for the rate of noncompliance of an internal control attribute" (5:304). This procedure, called finite Bayesian procedure (FBP) "correctly assumes a finite population, but in addition it exploits the auditor's prior information through a prior distribution" (5:304). However, previous research has indicated that "prior assessment provides the biggest stumbling block to the use of Bayesian methods" (5:313).

Deloitte, Haskins, and Sells (DH&S) developed another Bayesian approach using DUS. McCray submits the following analysis of the DH&S method:

> This approach [Warren, 1979] provides a formula for calculating the ultimate risk of failing to detect a material error in an account. However, the formula is seriously deficient for two reasons. First, one can construct numerous counter-examples which demonstrate that the actual ultimate risk can be at least four times larger than the risk calculated using the DH&S approach [Kinney, 1983]. Second, the DH&S formula does not take into consideration any measure of the

auditor's degree of uncertainty assigned to the evaluation of internal accounting control or sample information [10:36].

It is for these two reasons that the only value that can be calculated is "the probability of failing to detect a material error....No measure of the 'ultimate expectation' (posterior expectation) of the amount of total error in the account is possible" (10:36).

In the same research McCray proposed another Bayesian method

> ...that generates a discrete posterior probability distribution on the expected total error in a population for any dollar unit sample and any given discrete or continuous prior probability distribution on the expected total error in a population. The model can be used with any sample size and with any number of overstatements and understatements [10:35].

In analyzing his model, McCray comes to the conclusion that his method works as predicted and that comparing the bounds of his method with the multinomial bound it appears to be "...an acceptable model for evaluating dollar unit samples even if an informative prior probability distribution on the total expected error is not available" (10:50). However, further research is required before stronger conclusions can be made about the potential of the McCray method.

Another Bayesian based model proposed by Cox and Snell in 1979 was examined by Godfrey and Neter. The study, completed in 1984, was accomplished in order to

> ...extend the limited research performed to date on constructing Bayesian bounds for the population total error amount by (1) studying the characteristics of the Bayesian bound proposed by Cox and Snell [1979], (2) developing alternative Bayesian models in which the Cox and Snell assumptions are modified to see how sensitive the bounds are to these assumptions, and (3) conducting a simulation study to assess the robustness of the Cox and Snell bound to a wide variety of population conditions [7:3].

After clearly defining and discussing the Cox and Snell model, Godfrey and Neter conducted a simulation study along with four modified versions of the Cox and Snell model. They came to the following conclusions based on 21 study populations of sample size 100:

> [1] The Cox and Snell... bounds are substantially affected by the prior parameters [of error rate and mean taint]...
>
> [2] the Cox and Snell bound is: (1) robust, i.e, provides high coverage for all populations, (2) highly efficient compared to the Stringer bound for populations whose total error amount is low, and (3) moderately efficient for populations whose total error amount is large [7:35-36].

Conclusion

It is evident that predicting upper error limits for accounting populations with a certain level of confidence has been a major problem to auditors. Efforts to address this problem have been made, but the there are still many problems in developing a method to predict the upper error bound. It should also be obvious from the literature review that more research is necessary as well as justified to resolve the problem of confidence levels that are conservative.

II. Methodology

The research methodology consisted of five steps. The first was the collection of the data that was necessary to accomplish the research objectives. The second step consisted of writing a computer program (appendix A) to randomly select those line items which were to be seeded with error. The third step was to revise a computer program (appendix B) that would simulate the DUS-cell method as well as allocate error to those line items that were identified in step two. The fourth step was to revise a computer program (appendix C) to compute the coverage and relative tightness. Finally, the fifth step entailed a statistical analysis using the Statistical Analysis System (SAS) on the Boeing Time Share System, of the results, including a synopsis by error rate, error clustering, mean taint, and error amount intensity.

The Models

DUS-cell Bound. The DUS-cell bound considers the "whole sequence of population dollars subdivided into a number of equal cells, each cell width in dollars being equal to the average sampling interval" (6:135). The average sampling interval is calculated by dividing the total dollar amount of the population by the size of the sample. For example, consider an accounting population with a total dollar amount of $1,000,000 and a sample of 100 is

to be drawn, then the average sampling interval would be \$10,000 (\$1,000,000/100). After the average sampling interval is determined, one dollar is randomly selected from each cell and "the physical unit containing each selected dollar is audited, and the tainting of the dollar computed" (6:135). After all the taints have been computed they are ranked in order of decreasing value.

For each taint identified an upper error limit (UEL) factor is computed based on the Poisson approximation to the Binomial distribution. The UEL factor consist of three components; basic precision (BP), most likely error (MLE), and precision gap widening (PGW). Basic precision establishes a floor at which an auditor would expect that no errors would be found (6:127). Each sample error will raise the upper error limit from this value. For a confidence level of 95 percent, the value of BP has been identified as 3.00 and for an 85 percent confidence level, the BP value is 1.90. The most likely error is "a projection of the sample error value rate found", and will be equal to the number of errors found (6:127). The final component is the PGW which "represents the amount by which the total precision gap has increased as a result of finding" n tainted errors. (6:127). The UEL factor is a summation of the three components. For example,

Error stage	BP	MLE	PGW	UEL
0	3.00	-	-	3.00
1		1.00	.75	4.75
2		1.00	.55	6.30
3		1.00	.46	7.76
4		1.00	.40	9.16
5		1.00	.36	10.52
	----	----	----	
	3.00	5.00	2.52	

(6:125)

Figure 2.1. UEL Factor Computation

However, a table has been developed by Leslie, Teitlebaum, and Anderson which indicates the UEL factor for the corresponding number of errors.

Once the UEL factors have been listed and the taints have been listed in decending order two computation are then accomplished. The first one is the 'load and spread' value which is calculated by adding the UEL of the previous stage and the taint. The other is the 'simple spread' value which is computed by multiplying the UEL factor by the cumulative average taint of the errors found except at stage 0 where it is the just the UEL factor itself. For example, assume that a "sample of 100 drawn from a population of $1,000,000 (average sampling interval $10,000) yields five taintings: 40%, 10%, 80%, 90%, and 30%" (6:142). The upper error limit at a 95% confidence level would be:

A Error stage	B UEL factor	C Taint- ings	D Cum avg tainting	E UEL of previous stage (H)	F Load and Spread E+C	G Simple Spread BxD	H Stage UEL max (F,G)
0	3.00	-	-	-	-	-	3.000
1	4.75	.90	.90	3.000	3.900	4.275	4.275
2	6.30	.80	.85	4.275	5.075	5.355	5.355
3	7.76	.40	.70	5.355	5.755	5.432	5.755
4	9.16	.30	.60	5.755	6.055	5.496	6.055
5	10.52	.10	.50	6.055	6.155	5.260	6.155

(6:142)

Figure 2.2. Example of DUS-cell Method (95% Confidence Level)

$$
\begin{aligned}
\text{UEL} &= \text{Stage UEL} \times \text{Average Sampling Interval} \qquad (1)\\
&= 6.155 \times \$10{,}000\\
&= \$61{,}550
\end{aligned}
$$

<u>Modified DUS-cell Bound</u>. The modified DUS-cell method uses the same procedure as the DUS-cell to calculate the stage UEL. However, instead of projecting the stage UEL over the average sampling interval, as done in the DUS-cell, the modified DUS-cell projects the stage UEL over the average dollar amount not sampled. In other words, the book value dollar amount of those items sampled are removed from the total dollar amount of the population, then the stage UEL is applied to the average dollar value not sampled. After this calculation is complete the dollar error amount identified in the sample is added to the projected dollar error. For example, consider the example above where a sample of 100 was drawn from a population of $1,000,000 which yielded five error with taints of 40%, 10%, 80%, 90%,

and 30%. Also, assume that the total book value of the sample was $100,000 and that the book values of those five samples in error were $500, $700, $1,000, $1,300, and $1,500 respectively. Applying the modified DUS-cell method the following UEL would be attained:

A Error	B Taint	C Audited Value	D Book Value	E Error Amount (C-D)
1	.40	$ 700	$ 500	$ 200
2	.10	770	700	70
3	.80	1800	1000	800
4	.90	2470	1300	1170
5	.30	1950	1500	450
Total			5000	2690

Figure 2.3. Example of Modified DUS-cell Method (95% Confidence Level)

```
UEL = (Ty-sampbv)/n * Stage UEL + samper          (2)
    = ($1,000,000-$100,000)/100*6.155 + $2690
    = $58,085
```

where:

- Ty = total book value of the population
- n = sample size
- sampbv = total book value of sample
- samper = total amount of error in sample
- Stage UEL = same as DUS-cell Evaluation

<u>Stringer Bound</u>. The Stringer method was first introduced by Kenneth W. Stringer in 1963 and uses a Poisson approximation to the Binomial distribution. The Stringer bound makes use of three components in obtaining an upper

error limit for an accounting population. These three components are basic precision (BP), most likely error (MLE), and precision gap widening (PGW). Although these have identical nominclature as the three components used to compute the UEL factor of the DUS-cell bound, the method for calculating the MLE and PGW is accomplished differently. BP on the other hand, establishes a floor at which the auditor would intuitively expect that no errors would be found, which is identical to the BP description for the DUS-cell method. The MLE is calculated by summing the taints for each of the errors observed in the sample. The PGW is the summation of the products of the taints and the precision gap widening factors. The taints must be ordered in decreasing value just as they were for the DUS-cell method. Applying the Stringer method to the example described above in the DUS-cell method, results in the following UEL:

A Error stage	B Taint-ings	C PGW factor	D BP	E MLE +B	F PGW BxC	G Stage UEL D+E+F
0	-		3.00	-	-	3.000
1	.90	.75		.90	.675	4.575
2	.80	.55		1.70	1.115	5.815
3	.40	.46		2.10	1.299	6.399
4	.30	.40		2.40	1.419	6.819
5	.10	.36		2.50	1.455	6.955

Figure 2.4. Example of the Stringer Method (95% Confidence Level)

UEL = Stage UEL x Average Sampling Interval (1)
= 6.955 x \$10,000
= \$69,550

Modified Stringer Bound. The modified Stringer method uses the same technique as the Stringer bound to compute the stage UEL. However, instead of projecting the stage UEL over the average sample interval, as done in the Stringer bound, the modified Stringer projects the stage UEL over the average dollar amount not sampled and then adds the error amount in the sample on to the projection amount to arrive at the UEL. Applying the modified Stringer bound to the example used in the modified DUS-cell method where a sample of 100 is drawn with a total book value of \$100,000 from a population of \$1,000,000 where five errors were present with taintings of 40%, 10%, 80%, 90%, and 30% with corresponding book values of \$500, \$700, \$1,000, \$1,300, and \$1,500, yields the following UEL:

A Error	B Taint	C Audited Value	D Book Value	E Error Amount (C-D)
1	.4	\$ 700	\$ 500	\$ 200
2	.1	770	700	70
3	.8	1800	1000	800
4	.9	2470	1300	1170
5	.3	1950	1500	450
Total			5000	2690

Figure 2.5 Example of the Modified Stringer Method
(95% Confidence level)

$$\begin{aligned} UEL &= (Ty - sampbv)/n * \text{Stage } UEL + samper \qquad (3) \\ &= (\$1{,}000{,}000 - \$100{,}000)/100 * 6.955 + \$2690 \\ &= \$65{,}285 \end{aligned}$$

where:

Ty	=	total book value of the population
n	=	sample size
sampbv	=	total book value of sample
samper	=	total amount of error in sample
Stage UEL	=	same as the Stringer method

The reader should note that the only difference between the UEL calculation for the modified Stringer bound and the modified DUS-cell bound is stage UEL which was computed using the procedure for the Stringer bound and DUS-cell bound respectively.

Data Collection

The accounting population used to accomplish the research objectives was one of those included in Neter and Loebbecki (11:21-24). The population is comprised of 7026 accounts receivable from a manufacturing company. All account balances greater than $100,000 were removed on the premise that these accounts would be audited on a 100 percent basis. The dollar values in the population range from $.10 to $98,162.70 with a total dollar amount of $13,671,503.00, a mean book value of $1,945.84, a median of $6,835,751.50, a standard deviation of $7,021.61, a skewness of 7.94 and a kurtosis of 78.17.

Sample Generation

In order to analyze the performance of the upper error limit bounds, it was first necessary to identify line items to be seeded with error. In accomplishing this task, a computer program (appendix A) was used to randomly identify those line items that were seeded with error. However, with the research objectives in mind it was necessary to first find the median dollar amount and its corresponding line item. Then 80 percent of those line items below and 80 percent of those line items above the line item containing the median book value were randomly selected to be in error. The reason that 80 percent of the line items were selected from each stratum (high and low) was because the anticipated distribution of errors could possible cause that percent of the errors to occur in either the upper or lower stratum.

Error Allocation

For the purpose of evaluating the, error allocation was accomplished on three dimensions; error rates, error clusterings, and mean taints. Four different error rates, .50, .30, .15, and .01, were used to represent accounting populations with high (.50), medium (.30 and .15) and low (.01) line error rates. Error clustering is the distribution of the errors; that is, the proportion or errors seeded in each stratum. For the purposes of this analysis three categories of error clustering were used. They were uniform, increasing and decreasing which contained

a 1 to 1 ratio, 1 to 2 ratio, and a 2 to 1 ratio, respectively. Error taints were separated into two categories, high and low. The error taint was allotted based on the location of the error. That is, for high error taints a value of .4 was used for the lower stratum and a value of .2 for the upper stratum. When error taints were low, a value of .2 was used for the lower stratum and a value of .1 was used for the upper stratum.

Consequently, it was necessary to create 24 (4x3x2) study populations, each containing some combination of error rate, error clustering, and mean taint. Each population was then sampled 500 times with a sample size of 200 to measure the performance ability of the four bounds. Because of the problems encountered when using classical sampling techniques on accounting populations a method known as Dollar Unit Sampling (DUS) was used. Neter, Leitch, and Fienburg provided and excellent description and illustration when they stated:

> In Dollar Unit Sampling, the sample unit is the individual dollar. Once a sample dollar has been selected, the audit unit (such as an invoice or an account receivable) to which the sample dollar belongs is identified and audited. The error found in the audit unit is then prorated to each of the individual dollars in the unit and the sample dollar receives this prorated error amount.
>
> For example, consider the simple illustration in table 1 where the population consists of five accounts. The sampling procedure is as follows:
>
> 1) The auditor determines cumulative book amount ranges for all audit units.
> 2) The cumulative book total, X, is the total number of dollar units in the population. In this example, the total number of dollar units in the population is 500.
> 3) To select a sample of, say, three dollar units, the

auditor chooses three random numbers from 1 to X (500 in this example). These random numbers are illustrated in the last column of Table 1.

4) As shown in Table 1, audit units A, D, and E are selected for audit. The auditor examines the entire audit unit to which the sample dollar belongs.

5) He then prorates the total error for the audit unit to each dollar in the unit. Suppose the audit of A discloses a $20 overstatement error. The sample dollar unit from this audit unit is then assumed to have an overstatement error of 20/100 or $.20 [3:79].

All items with a greater dollar amount than the average sampling interval were audited on a 100% basis, and the error amount was added to the upper error limit bound.

Table 2.1

Dollar Unit Sampling Illustration

Account (Audit Unit)	Book Value	Cumulative Book Amount Range	Random Numbers
A	$100	1-100	39
B	50	101-150	
C	20	151-170	
D	200	171-370	241
E	130	371-500	486

(3:79)

Once a DUS sample was generated an upper error limit bound was calculated for DUS-cell, modified DUS-cell, Stringer, and modified Stringer bounds at both the 85 and 95 percent level of confidence. Therefore, there were eight upper error limits for each of the study populations. Analysis of the upper error limits focused on the coverage of each of the bounds and how the coverage varied with

respect to the error rate, error clustering, mean taint, and error amount intensity. Furthermore, the analysis examined the relative tightness of each of the bounds with respect to the stringer bound at the same level of confidence. Also, it was important to analyze how relative tightness varied with respect to error rate, error clustering, mean taint, and error amount intensity.

III. Results and Analysis

This chapter contains the analysis of results of the research. The chapter includes four major sections; one for each research question. The first section examines the robustness of the Stringer, modified Stringer, DUS-cell, and modified DUS-cell bounds. The second section evaluates the tightness of the DUS-cell bounds compared to those of the Stringer bound. The third section looks at the effect that different levels of nominal confidence may have. Finally, the last section examines the possible effects of error rate, error clustering, mean taint, and error amount intensity on coverage and relative tightness. The following table is a list of acronyms that will be used throughout this chapter to denote specific bounds.

Table 3.1

Acronym Definitions

Unmodified Bounds		
S1	-	the Stringer bound, at 95% nominal confidence level
S3	-	the Stringer bound, at 85% nominal confidence level
C1	-	the DUS-cell bound, at 95% nominal confidence level
C3	-	the DUS-cell bound, at 85% nominal confidence level
Modified Bounds		
S2	-	the Stringer bound, at 95% nominal confidence level
S4	-	the Stringer bound, at 85% nominal confidence level
C2	-	the DUS-cell bound, at 95% nominal confidence level
C4	-	the DUS-cell bound, at 85% nominal confidence level

Research Question One

The first research question addressed the robustness of the bounds. Specifically stated, are the Stringer, modified Stringer, DUS-cell, and modified DUS-cell bounds robust for the given study populations? The objective was to examine the coverage of the bounds to see if the actual levels of confidence were equal to or greater than the nominal confidence levels for all populations in the study. Tables 3.2a and 3.2b depict the respective coverages for both the unmodified and modified Stringer and DUS-cell bounds.

As tables 3.2a and 3.2b show the only two bounds that are robust are the unmodified Stringer bounds at both nominal levels of confidence (S1,S3). The other bounds are not robust. However, the unmodified DUS-cell bound at both confidence levels (C1,C3) and the modified Stringer bound at the 95 percent level of confidence (S2) just missed qualifying as robust bounds, as the actual coverage for each bound was less than the nominal level for just one study population. It is interesting to point out that all three bounds failed to cover dollar errors adequately in study population 4. The modified DUS-cell bound at both levels of confidence (C2,C4) failed six times with all six occurring when the line item error rate of the study population was 50 percent. The modified Stringer bound at 85 percent confidence (S4) failed to meet the nominal confidence level for three study populations. Again, like the modified DUS-

Table 3.2a

Coverage of Unmodified Bounds

Confidence Level	95%	85%	95%	85%
Study Population	S1	S3	C1	C3
1	1.0000	.9740	.9820	.9100
2	1.0000	.9940	.9700	.9160
3	1.0000	.9820	.9860	.9100
4	1.0000	.9520	.9440	.8180
5	.9980	.9520	.9600	.8840
6	1.0000	.9800	.9600	.8780
7	1.0000	.9960	.9940	.9080
8	1.0000	1.0000	1.0000	.9660
9	1.0000	1.0000	.9980	.9660
10	1.0000	.9980	.9960	.9440
11	1.0000	.9680	.9580	.8780
12	1.0000	1.0000	1.0000	.9760
13	1.0000	1.0000	1.0000	.9980
14	1.0000	1.0000	1.0000	.9980
15	1.0000	1.0000	1.0000	.9980
16	1.0000	1.0000	1.0000	1.0000
17	1.0000	1.0000	1.0000	1.0000
18	1.0000	1.0000	1.0000	1.0000
19	1.0000	1.0000	1.0000	1.0000
20	1.0000	1.0000	1.0000	1.0000
21	1.0000	1.0000	1.0000	1.0000
22	1.0000	1.0000	1.0000	1.0000
23	1.0000	1.0000	1.0000	1.0000
24	1.0000	1.0000	1.0000	1.0000
Mean Coverage	0.9999	0.9915	0.9895	0.9582

Table 3.2b

Coverage of Modified Bounds

Confidence Level	95%	85%	95%	85%
Study Population	S2	S4	C2	C4
1	.9700	.8340	.8640	.6920
2	.9980	.9480	.9260	.7600
3	.9720	.7860	.8180	.5840
4	.9000	.7380	.7260	.6060
5	.9740	.8860	.8940	.8360
6	1.0000	.9100	.8540	.6640
7	.9980	.9940	.9760	.8760
8	1.0000	1.0000	1.0000	.9880
9	1.0000	.9980	.9880	.9220
10	1.0000	.9860	.9820	.8860
11	1.0000	.9720	.9700	.9040
12	1.0000	1.0000	1.0000	.9840
13	1.0000	.9980	.9880	.9660
14	1.0000	1.0000	1.0000	.9960
15	1.0000	1.0000	1.0000	.9740
16	1.0000	.9980	1.0000	.9960
17	1.0000	1.0000	1.0000	1.0000
18	1.0000	1.0000	1.0000	1.0000
19	1.0000	1.0000	1.0000	1.0000
20	1.0000	1.0000	1.0000	1.0000
21	1.0000	1.0000	1.0000	1.0000
22	1.0000	1.0000	1.0000	1.0000
23	1.0000	1.0000	1.0000	1.0000
24	1.0000	1.0000	1.0000	1.0000
Mean Coverage	0.9922	0.9603	0.9557	0.9014

cell bounds, the failures occurred when the line item error rate was 50 percent. Also, two of the three study populations for which the modified Stringer bound at the 85 percent nominal confidence level (S4) failed to achieve adequate coverage had high mean taints of .4/.2.

Research Question Two

The second research question addressed the relative tightness of the bounds. The objective was to determine if the DUS-cell bound yields a tighter bound than the corresponding Stringer bound. The relative tightness was computed by summing over 500 replications, the quotient of a particular Stringer bound divided by the corresponding DUS-cell bound and then dividing the resulting sum by the 500 replications. Furthermore, to compare the relative tightness of the modified Stringer bound with the unmodified Stringer bound the former was divided into the latter. However, to compare the unmodified DUS-cell bound with the modified DUS-cell bound a calculation that must be accomplished is to multiply the relative tightness of the modified Stringer by the corresponding modified DUS-cell bound. This provides a common baseline, that is, the unmodified Stringer bound for comparisons between the unmodified and modified DUS-cell bounds. As tables 3.3a and 3.3b show, the DUS-cell bound is tighter than the corresponding Stringer bound in all cases. At both levels of confidence the mean relative tightness of the unmodified

Table 3.3a

Relative Tightness for 95% Nominal Confidence Level bounds

Study Population	S1	C1	S2	C2
1	1.0000	1.0839	1.1216	1.0685
2	1.0000	1.1230	1.1005	1.0978
3	1.0000	1.0856	1.1406	1.0712
4	1.0000	1.1230	1.1572	1.1034
5	1.0000	1.0756	1.0769	1.0593
6	1.0000	1.1194	1.1052	1.0954
7	1.0000	1.1298	1.0763	1.1006
8	1.0000	1.1562	1.0600	1.1183
9	1.0000	1.1516	1.1158	1.1217
10	1.0000	1.1932	1.1323	1.1566
11	1.0000	1.0986	.9882	1.0703
12	1.0000	1.1841	1.0634	1.1390
13	1.0000	1.2010	1.1056	1.1584
14	1.0000	1.2118	1.1102	1.1675
15	1.0000	1.2050	1.1360	1.1665
16	1.0000	1.1660	1.2303	1.1476
17	1.0000	1.2227	1.0902	1.1719
18	1.0000	1.2200	1.0710	1.1667
19	1.0000	1.2113	1.1348	1.1713
20	1.0000	1.1821	1.1769	1.1543
21	1.0000	1.1420	1.2363	1.1275
22	1.0000	1.1040	1.2733	1.0953
23	1.0000	1.0321	1.3348	1.0316
24	1.0000	1.0382	1.3294	1.0374
Mean Relative Tightness	1.0000	1.1403	1.1442	1.1166

Table 3.3b

Relative Tightness for 85% Nominal Confidence Level bounds

Study Population	S3	C3	S4	C4
1	1.0000	1.0567	1.1023	1.0457
2	1.0000	1.0829	1.0786	1.0650
3	1.0000	1.0578	1.1229	1.0475
4	1.0000	1.0836	1.1379	1.0694
5	1.0000	1.0508	1.0579	1.0393
6	1.0000	1.0800	1.0850	1.0631
7	1.0000	1.0883	1.0504	1.0672
8	1.0000	1.1072	1.0298	1.0795
9	1.0000	1.1060	1.0869	1.0833
10	1.0000	1.1374	1.1004	1.1089
11	1.0000	1.0671	.9599	1.0468
12	1.0000	1.1292	1.0273	1.0950
13	1.0000	1.1440	1.0689	1.1104
14	1.0000	1.1550	1.0699	1.1188
15	1.0000	1.1544	1.0943	1.1212
16	1.0000	1.1377	1.1948	1.1189
17	1.0000	1.1633	1.0464	1.1218
18	1.0000	1.1603	1.0264	1.1172
19	1.0000	1.1650	1.0857	1.1282
20	1.0000	1.1507	1.1270	1.1223
21	1.0000	1.1240	1.1954	1.1075
22	1.0000	1.0939	1.2430	1.0831
23	1.0000	1.0314	1.3260	1.0307
24	1.0000	1.0371	1.3179	1.0361
Mean Relative Tightness	1.0000	1.1098	1.1068	1.0845

DUS-cell bound is almost identical to the mean relative tightness of the modified Stringer bound. At the 95 percent nominal confidence level, the mean relative tightness of the unmodified DUS-cell bound is 1.1403 and the mean relative tightness of the modified Stringer bound is 1.1442, a difference of only .0039. At the 85 percent nominal confidence level, the mean relative tightnesses of the unmodified DUS-cell bound and the modified Stringer bound are 1.1098 and 1.1068, respectively; a difference of .0030. Study population eleven contained a Stringer bound that was tignter than the modified Stringer bound at both levels of confidence. This is an interesting result since the modified bound was assumed to be less conservative than the unmodified version of the bound. The error distribution of study population 11 was a line item error rate of 30 percent, a 1:2 error clustering ratio, and a mean taint of .4/.2.

Research Question Three

Research question three addressed the effect that the nominal levels of confidence had on the robustness of the DUS-cell bounds. The purpose for examining the possible effect of different confidence levels is that the DUS-cell bound may not perform the same under assumptions of greater risk (lower nominal confidence). To accomplish this objective the mean coverages, for all 24 study populations, of the modified and unmodified DUS-cell bounds at both

nominal confidence levels were used to compute the average percentage above the nominal confidence level of each DUS-cell bound. To normalize the data, the percentage above the nominal confidence was based on the possible increase each bound could make. For example, at a nominal confidence level of 85 percent the bound could increase by as much as 15 percent, but at the 95 percent nominal confidence level, it could only increase by 5 percent at most. Therefore, in order to compare the possible effect of different nominal confidence levels, the nominal confidence levels were subtracted from the actual mean coverage of each DUS-cell bound with the result being divided by the possible increase. For example, consider the mean coverage for the DUS-cell at 95 percent nominal confidence level (C1) which was 98.95 percent. To compute the average percentage above the nominal confidence level, 95 percent was subtracted from 98.95 percent and the result was then divided by 5 percent (100%-95%) to yield an average percentage above the nominal confidence level of 79 percent. This would indicate that on the average the DUS-cell bound at 95 percent nominal confidence produced an actual coverage that was 79 percent closer to 100 percent than is required. Figure 3.1 shows the average percentage increase in coverage above the nominal confidence level for all four DUS-cell bounds.

On the average the percentage increase for the unmodified DUS-cell bounds (C1,C3) was not significantly

A DUS-cell Bound	B Actual Confidence Level	C Nominal Confidence Level	D Differ-ence (B-C)	E Possible Increase (1.00-B)	F Avg % Increase (D/E)
C1	.9895	.95	.0395	.05	.79
C2	.9582	.95	.0082	.05	.16
C3	.9557	.85	.1057	.15	.70
C4	.9014	.85	.0514	.15	.34

Figure 3.1 Average Percentage Increase Above the Nominal Confidence Level

different at the two levels of nominal confidence. However, the average percentage increase was more than twice as much for the modified DUS-cell at 85 percent of nominal confidence level (C4) than the modified DUS-cell at 95 percent nominal confidence level (C2). This indicates that the level of nominal confidence specified by the auditor can affect the performance of the modified DUS-cell bounds.

Research Question Four

The fourth research question was which characteristics of accounting populations have a significant impact on the performance of the DUS-cell method in calculating the upper error limit. The objective was to find out if error rate, error clustering, mean taint, and error amount intensity (EAI) have an impact on the performance of the DUS-cell bound. The technique used to accomplish this objective was to sort the study populations by line item error rate, examine the performance of the DUS-cell bound at the different error rates, and determine what affects, if any,

the different error rates had on the performance of the DUS-cell bound. The same technique was used for error clustering, mean taint, and EAI to see what affect each of them may have had on the performance of the DUS-cell bound. The performance measures that were used were coverage and relative tightness. Table 3.4 shows the study populations and the error distributions that each contained.

Line Item Error Rate Effects on Coverage. Table 3-5 shows the maximum value, minimum value, mean, and standard deviation of the coverage of each bound by line item error rate. With a line item error rate of 1 percent, all eight bounds were extremely conservative. All eight bounds had an actual coverage of 100 percent.

Even those populations with a 15 percent line item error rate were provided high coverage with all eight bounds. The unmodified Stringer bound at both levels of nominal confidence (S1,S3), the modified Stringer bound at 95 percent nominal confidence (S2), and the unmodified DUS-cell bound at 95 percent confidence level (C1) continued to be extremely conservative. All four bounds covered dollar errors at the 100 percent level. The remaining four bounds, (S4,C2,C3,C4) also proved to be highly conservative but not at the 100 percent level. The worst performer was the modified DUS-cell at 85 percent nominal confidence (C4) at 96.6 percent which is far above the nominal confidence level of 85 percent. Even at the 15 percent line item error rate

Table 3.4

Study Populations Error Distributions

Study Popu- lations	Line Item Error Rate	Error Clust- ering	Error Taint	EAI
1	50%	1:1	.4/.2	.126925
2	50%	1:1	.2/.1	.099613
3	50%	2:1	.4/.2	.134320
4	50%	2:1	.2/.1	.091496
5	50%	1:2	.4/.2	.167761
6	50%	1:2	.2/.1	.109730
7	30%	1:1	.4/.2	.086705
8	30%	1:1	.2/.1	.067388
9	30%	2:1	.4/.2	.056980
10	30%	2:1	.2/.1	.039991
11	30%	1:2	.4/.2	.098856
12	30%	1:2	.2/.1	.045309
13	15%	1:1	.4/.2	.033845
14	15%	1:1	.2/.1	.026071
15	15%	2:1	.4/.2	.020810
16	15%	2:1	.2/.1	.008376
17	15%	1:2	.4/.2	.026538
18	15%	1:2	.2/.1	.027557
19	1%	1:1	.4/.2	.014291
20	1%	1:1	.2/.1	.008675
21	1%	2:1	.4/.2	.005027
22	1%	2:1	.2/.1	.002798
23	1%	1:2	.4/.2	.000646
24	1%	1:2	.2/.1	.000846

Table 3.5

Coverage of Bounds by Error Rate

BOUND	N	MAXIMUM VALUE	MINIMUM VALUE	MEAN	STANDARD DEVIATION
Error Rate of 50%					
S1	6	1.00000000	0.99800000	0.99966667	0.00081650
S2	6	1.00000000	0.90000000	0.96900000	0.03632630
S3	6	0.99400000	0.95200000	0.97233333	0.01703721
S4	6	0.94800000	0.73800000	0.85033333	0.07924056
C1	6	0.98600000	0.94400000	0.96700000	0.01563330
C2	6	0.92600000	0.72600000	0.84700000	0.06967065
C3	6	0.91600000	0.81800000	0.88600000	0.03672601
C4	6	0.83600000	0.58400000	0.69033333	0.09506559
Error Rate of 30%					
S1	6	1.00000000	1.00000000	1.00000000	0.00000000
S2	6	1.00000000	0.99800000	0.99966667	0.00081650
S3	6	1.00000000	0.96800000	0.99366667	0.01267544
S4	6	1.00000000	0.97200000	0.99166667	0.01098484
C1	6	1.00000000	0.95800000	0.99100000	0.01633401
C2	6	1.00000000	0.97000000	0.98600000	0.01239355
C3	6	0.97600000	0.87800000	0.93966667	0.03881065
C4	6	0.98800000	0.87600000	0.92666667	0.04859081
Error Rate of 15%					
S1	6	1.00000000	1.00000000	1.00000000	0.00000000
S2	6	1.00000000	1.00000000	1.00000000	0.00000000
S3	6	1.00000000	1.00000000	1.00000000	0.00000000
S4	6	1.00000000	0.99800000	0.99933333	0.00103280
C1	6	1.00000000	1.00000000	1.00000000	0.00000000
C2	6	1.00000000	0.99800000	0.99966667	0.00081650
C3	6	1.00000000	0.98800000	0.99733333	0.00467618
C4	6	1.00000000	0.96600000	0.98866667	0.01478738
Error Rate of 1%					
S1	6	1.00000000	1.00000000	1.00000000	0.00000000
S2	6	1.00000000	1.00000000	1.00000000	0.00000000
S3	6	1.00000000	1.00000000	1.00000000	0.00000000
S4	6	1.00000000	1.00000000	1.00000000	0.00000000
C1	6	1.00000000	1.00000000	1.00000000	0.00000000
C2	6	1.00000000	1.00000000	1.00000000	0.00000000
C3	6	1.00000000	1.00000000	1.00000000	0.00000000
C4	6	1.00000000	1.00000000	1.00000000	0.00000000

all the bounds continued to perform very conservatively. In addition, the mean of all DUS-cell bounds were equal to or less than the corresponding Stringer bound. This indicates that at the 15 percent line item error rate, the Stringer bounds are slightly more conservative than the DUS-cell bounds.

As the line item error rate increased to 30 percent, the coverage of all eight bounds moved closer to the nominal confidence levels. However, on the average, the DUS-cell bounds moved closer to the nominal confidence levels faster than the corresponding Stringer bounds. This indicates that, as the error rate increases, the DUS-cell bounds adjust faster than the Stringer bounds. All eight bounds performed satisfactorily for all study populations with actual coverages at or above the nominal confidence levels.

The 50 percent line item error rate was a problem for some of the bounds. For instance, the modified DUS-cell bound at both nominal confidence levels (C2,C4) failed to provide adequate coverage for all six of the study populations containing a 50 percent line item error rate. The only two bounds that were robust were the unmodified Stringer bounds at both nominal confidence levels (S1,S3). It was also at this error rate that the modified Stringer at 95 percent nominal confidence (S2) and the unmodified DUS-cell at both levels of confidence (C1,C3) failed to provide adequate coverage for one population disqualifying them as

robust bounds. The modified Stringer bound at 85 percent nominal confidence (S4) failed to provide sufficient coverage three times (50 percent of the time). However, on the average, the only two bounds to drop below the nominal confidence levels were the modified DUS-cell bounds (C2,C4). Therefore, the 50 percent line item error rate affected the modified DUS-cell bounds significantly more than the unmodified DUS-cell bounds and the modified Stringer bounds. Mean coverages for all the bounds approached the nominal level with coverages for the modified DUS-cell bound at both nominal confidence levels (C2,C4) actually falling below the nominal levels.

Error Clustering Effects on Coverage. To evaluate the possible effects of error clustering on coverage, the study populations were sorted by error clustering ratios. Table 3.6 shows the maximum value, minimum value, mean ,and standard deviation for the coverage of each bound for the various clusterings of errors. Although all three groups of clusterings contain study populations with bounds that fail to provide sufficient coverage, a closer look reveals that a clustering ratio of 2:1 has a greater effect on the performance of the bounds. For example, if all eight bounds, both the modified and unmodified, Stringer and DUS-cell bounds, at both nominal confidence levels, are considered for all 24 study populations, there were 18 incidents where the coverage fell below the desired nominal

Table 3.6

Coverage of Bounds by Error Clustering

BOUND	N	MAXIMUM VALUE	MINIMUM VALUE	MEAN	STANDARD DEVIATION
Error Clustering Ratio 1:1					
S1	8	1.00000000	1.00000000	1.00000000	0.00000000
S2	8	1.00000000	0.97000000	0.99575000	0.01044373
S3	8	1.00000000	0.97400000	0.99550000	0.00899206
S4	8	1.00000000	0.83400000	0.97175000	0.05845083
C1	8	1.00000000	0.97000000	0.99325000	0.01131055
C2	8	1.00000000	0.86400000	0.97050000	0.05012841
C3	8	1.00000000	0.90800000	0.96075000	0.04242557
C4	8	1.00000000	0.69200000	0.90975000	0.12188724
Error Clustering Ratio 2:1					
S1	8	1.00000000	1.00000000	1.00000000	0.00000000
S2	8	1.00000000	0.90000000	0.98400000	0.03532704
S3	8	1.00000000	0.95200000	0.99150000	0.01712976
S4	8	1.00000000	0.73800000	0.93825000	0.10963543
C1	8	1.00000000	0.94400000	0.99050000	0.01938335
C2	8	1.00000000	0.72600000	0.93925000	0.10633068
C3	8	1.00000000	0.81800000	0.95450000	0.06431840
C4	8	1.00000000	0.58400000	0.87100000	0.17522883
Error Clustering Ratio 1:2					
S1	8	1.00000000	0.99800000	0.99975000	0.00070711
S2	8	1.00000000	0.97400000	0.99675000	0.00919239
S3	8	1.00000000	0.95200000	0.98750000	0.01881489
S4	8	1.00000000	0.88600000	0.97100000	0.04652495
C1	8	1.00000000	0.95800000	0.98475000	0.02105605
C2	8	1.00000000	0.85400000	0.96475000	0.05795503
C3	8	1.00000000	0.87800000	0.95200000	0.06019967
C4	8	1.00000000	0.66400000	0.92350000	0.12097579

confidence level. Those 18 incidents include 6 failures of the modified DUS-cell bound at 95 percent nominal confidence, 6 failures of the modified DUS-cell bound at 85 percent nominal confidence, 3 failures of the modified Stringer bound at 85 percent nominal confidence, 1 failure each for the DUS-cell bound at both nominal confidence levels and the modified Stringer bound at 95 percent nominal confidence. Of those 18 insufficient upper error limits 9, or 50 percent, occur in study populations which contain error with a 2:1 ratio in the lower and upper strata, respectively. Also, the 2:1 ratio clustering is the only one that results in any of the mean coverages dropping below the nominal confidence level. The 2:1 ratio clustering is also an error distribution characteristic of study population 4 which is the one population for which the unmodified DUS-cell bound at both levels of nominal confidence (C1,C3) and the modified Stringer bound at 95 percent nominal confidence(S2) failed to provide adequate coverage, disqualifying them as robust bounds.

Mean Taint Effects on Coverage. To evaluate the effect that mean taint had on coverage the study populations were sorted by mean taint. Table 3.7 shows the maximum value, minimum value, mean, and standard deviation for the coverage of each bound for the two mean taints. There was little effect of mean taint on coverage. Of the 18 coverage failures, there was almost an even split between the taints

Table 3.7

Coverage of Bounds by Mean Taint

BOUND	N	MAXIMUM VALUE	MINIMUM VALUE	MEAN	STANDARD DEVIATION
Mean Taint of .4/.2					
S1	12	1.00000000	0.99800000	0.99983333	0.00057735
S2	12	1.00000000	0.97000000	0.99283333	0.01260471
S3	12	1.00000000	0.95200000	0.98933333	0.01645563
S4	12	1.00000000	0.78600000	0.95566667	0.07603269
C1	12	1.00000000	0.95800000	0.98983333	0.01561953
C2	12	1.00000000	0.81800000	0.95900000	0.06346653
C3	12	1.00000000	0.87800000	0.95350000	0.05077669
C4	12	1.00000000	0.58400000	0.89616667	0.13414635
Mean Taint of .2/.1					
S1	12	1.00000000	1.00000000	1.00000000	0.00000000
S2	12	1.00000000	0.90000000	0.99150000	0.02882076
S3	12	1.00000000	0.95200000	0.99366667	0.01434214
S4	12	1.00000000	0.73800000	0.96500000	0.07683986
C1	12	1.00000000	0.94400000	0.98916667	0.01964148
C2	12	1.00000000	0.72600000	0.95733333	0.08546486
C3	12	1.00000000	0.81800000	0.95800000	0.05936635
C4	12	1.00000000	0.60600000	0.90666667	0.14604690

with eight failures occurring with high mean taints and 10 failures occurring with low mean taints. Furthermore, the mean coverages of the bounds for study populations with high mean taints differ very little from coverages of corresponding bounds for study populations with low mean taints. Overall, the mean taints had very little effect on the coverage of the Stringer and DUS-cell bounds.

<u>Error Amount Intensity (EAI) Effects on Coverage</u>. The error amount intensity is a population characteristic and is calculated by dividing the total population dollar error by the total population book value. To evaluate the possible effects of error amount intensity on the coverage of the bounds, error amount intensities were ranked from low to high as shown in table 3.8. Natural breaks seemed to occur between observations 8 and 9 and again between observations 16 and 17 which means the error amount intensity was divided into three groups; low, medium, and high. The performance of the bounds within each group was then analyzed. Table 3.9 shows the maximum value, minimum value, mean, and standard deviation for the coverage of each bound within the three groupings by error amount intensity.

When populations have low error amount intensities all eight bounds were robust. However, all eight bounds were extremely conservative with a minimum coverage of 97.4 percent for an 85 percent nominal confidence level and a minimum coverage of 100 percent for a 95 percent nominal

Table 3.8

Ranked Order of Error Amount Intensity

Observation Number	Error Amount Intensity	Study Population	
1	0.000646	23	
2	0.000846	24	
3	0.002798	22	L
4	0.005027	21	O
5	0.008376	16	W
6	0.008675	20	
7	0.014291	19	
8	0.020810	15	
9	0.026071	14	
10	0.026538	17	M
11	0.027557	18	E
12	0.033845	13	D
13	0.039991	10	I
14	0.045309	12	U
15	0.056980	9	M
16	0.067388	8	
17	0.086705	7	
18	0.091496	4	
19	0.098856	11	H
20	0.099613	2	I
21	0.109730	6	G
22	0.126925	1	H
23	0.134320	3	
24	0.167761	5	

Table 3.9

Coverage of Bounds by Error Amount Intensity(EAI)

BOUND	N	MAXIMUM VALUE	MINIMUM VALUE	MEAN	STANDARD DEVIATION
EAI is Low					
S1	8	1.00000000	1.00000000	1.00000000	0.00000000
S2	8	1.00000000	1.00000000	1.00000000	0.00000000
S3	8	1.00000000	1.00000000	1.00000000	0.00000000
S4	8	1.00000000	0.99800000	0.99975000	0.00070711
C1	8	1.00000000	1.00000000	1.00000000	0.00000000
C2	8	1.00000000	1.00000000	1.00000000	0.00000000
C3	8	1.00000000	0.99800000	0.99975000	0.00070711
C4	8	1.00000000	0.97400000	0.99625000	0.00909867
EAI is Medium					
S1	8	1.00000000	1.00000000	1.00000000	0.00000000
S2	8	1.00000000	1.00000000	1.00000000	0.00000000
S3	8	1.00000000	0.99800000	0.99975000	0.00070711
S4	8	1.00000000	0.98600000	0.99775000	0.00483292
C1	8	1.00000000	0.99600000	0.99925000	0.00148805
C2	8	1.00000000	0.98200000	0.99600000	0.00701020
C3	8	1.00000000	0.94400000	0.97975000	0.02032416
C4	8	1.00000000	0.88600000	0.96775000	0.04197193
EAI is High					
S1	8	1.00000000	0.99800000	0.99975000	0.00070711
S2	8	1.00000000	0.90000000	0.97650000	0.03370036
S3	8	0.99600000	0.95200000	0.97475000	0.01683322
S4	8	0.99400000	0.73800000	0.88350000	0.09105571
C1	8	0.99400000	0.94400000	0.96925000	0.01686713
C2	8	0.97600000	0.72600000	0.87850000	0.08289580
C3	8	0.91600000	0.81800000	0.88775000	0.03222133
C4	8	0.90400000	0.58400000	0.74025000	0.12269562

level of confidence. As the error amount intensity increases, the bound conservativism tends to decline as demonstrated by the medium error amount intensity. On the average, all the means moved closer to the nominal levels of confidence with the exception of the unmodified Stringer and modified Stringer at the 95 percent nominal confidence level (S1,S2) which remained the same (100 coverage). Also, at the medium error amount intensity there were not any minimum coverages that dropped below the nominal confidence levels. However, when a population had a high error amount intensity, bound coverage was not so conservative. All 18 of the insufficient upper error limits with coverages less than the nominal levels fell into this category. Also, the mean coverages of the modified DUS-cell bounds at both levels of confidence ((C2,C4) fell below their respective nominal confidence levels. In general, error amount intensity had a significant impact on the coverage of the bounds. More specifically, a high error amount intensity tends to significantly reduce coverages of both modified DUS-cell bounds (C2,C4) and the modified Stringer at 85 percent nominal confidence (S4). High error amount intensity is a error distribution characteristic of study population 4, the only population for which both the unmodifed DUS-cell bounds (C1,C3) and the modified Stringer bound at 95 percent confidence (S2) failed to provide adequate coverage.

Line Item Error Rate Effects on Relative Tightness. To determine what effects the line item error rate had on relative tightness, the study populations were sorted by line item error rate and the relative tightness of the bounds for each error rate was examined. Table 3.10 lists the maximum value, minimum value, mean, and standard deviation of the relative tightness of each bound for the four line item error rates. The most noticeable effect that error rate has on relative tightness is at the 1 percent level. At this level that the greatest range and the highest standard deviation of relative tightness were encountered for all four of the DUS-cell bounds (C1,C2,C3,C4) and both the modified Stringer bounds (S2,S4). All four DUS-cell bounds (C1,C2,C3,C4) also produced their minimum values for relative tightness at the 1 percent line item error rate. However, the line item error rate at which all four DUS-cell bounds (C1,C2,C3,C4) were the tightest with respect to their corresponding Stringer bound was at 30 percent. The modified Stringer bounds (S2,S4) were the tightest with respect to the unmodified Stringer bounds at the 1 percent line item error rate. In general, all four DUS-cell bounds (C1,C2,C3,C4) were tighter than the corresponding Stringer bounds by greater amounts as the error rate decreases from 50 percent, through 30 percent, to 15 percent. However, the four DUS-cell bounds (C1,C2,C3,C4) decreased in relative tightness when moving from 15 a

Table 3.10

Relative Tightness of Bounds by Error Rate

BOUND	N	MAXIMUM VALUE	MINIMUM VALUE	MEAN	STANDARD DEVIATION
Error Rate of 50%					
S1	6	1.00000000	1.00000000	1.00000000	0.00000000
S2	6	1.15720000	1.07690000	1.11700000	0.02902985
S3	6	1.00000000	1.00000000	1.00000000	0.00000000
S4	6	1.13790000	1.05790000	1.09743333	0.02960390
C1	6	1.12300000	1.07560000	1.10175000	0.02226169
C2	6	1.10340000	1.05930000	1.08260000	0.01843464
C3	6	1.08360000	1.05080000	1.06863333	0.01506342
C4	6	1.06940000	1.03930000	1.05500000	0.01234666
Error Rate of 30%					
S1	6	1.00000000	1.00000000	1.00000000	0.00000000
S2	6	1.13230000	0.98820000	1.07266667	0.05061769
S3	6	1.00000000	1.00000000	1.00000000	0.00000000
S4	6	1.10040000	0.95990000	1.04245000	0.05020210
C1	6	1.19320000	1.09860000	1.15225000	0.03489893
C2	6	1.15660000	1.07030000	1.11775000	0.03007462
C3	6	1.13740000	1.06710000	1.10586667	0.02589221
C4	6	1.10890000	1.04680000	1.08011667	0.02162105
Error Rate of 15%					
S1	6	1.00000000	1.00000000	1.00000000	0.00000000
S2	6	1.23030000	1.07100000	1.12388333	0.05643412
S3	6	1.00000000	1.00000000	1.00000000	0.00000000
S4	6	1.19480000	1.02640000	1.08345000	0.05922711
C1	6	1.22270000	1.16600000	1.20441667	0.02058878
C2	6	1.17190000	1.14760000	1.16310000	0.00875968
C3	6	1.16330000	1.13770000	1.15245000	0.00978381
C4	6	1.12180000	1.11040000	1.11805000	0.00411133
Error Rate of 1%					
S1	6	1.00000000	1.00000000	1.00000000	0.00000000
S2	6	1.33480000	1.13480000	1.24758333	0.08100073
S3	6	1.00000000	1.00000000	1.00000000	0.00000000
S4	6	1.32600000	1.08570000	1.21583333	0.09852502
C1	6	1.21130000	1.03210000	1.11828333	0.07393520
C2	6	1.17130000	1.03160000	1.10290000	0.05891923
C3	6	1.16500000	1.03140000	1.10035000	0.05669274
C4	6	1.12820000	1.03070000	1.08465000	0.04267635

percent error rate to a 1 percent line item error rate.

Error Clustering Effects on Relative Tightness. To evaluate the possible effects of error clustering on relative tightness the study populations were sorted by error clustering and the relative tightness of the bounds was examined. Table 3.11 lists the maximum value, minimum value, mean, and standard deviation of relative tightness for each bound within the three categories of error clustering. The 1:2 error clustering ratio had the most noticeable effect on relative tightness; at this ratio level, the greatest range and highest standard deviation of relative tightness were found for both the modified and unmodified DUS-cell bounds at 95 percent nominal confidence level (C1,C2) as well as both modified Stringer bounds (S2,S4). The most stable error clustering ratio for both modified and unmodified DUS-cell bounds (C1,C2,C3,C4) was the 1:2 pattern. At this error clustering ratio, both the modified and unmodified DUS-cell bounds (C1,C2,C3,C4) had their smallest range and standard deviation for relative tightness. Similarly, the 1:1 error clustering ratio was the most stable for the modified Stringer bounds at both nominal confidence levels (S2,S4). At this error clustering ratio, the smallest ranges and standard deviation of relative tightness were found for these two bounds (S2,S4). Also at the 1:1 error clustering ratio, both the unmodified DUS-cell bounds (C1,C3) and the modified DUS-cell bound at

Table 3.11

Relative Tightness of Bounds by Error Clustering

BOUND	N	MAXIMUM VALUE	MINIMUM VALUE	MEAN	STANDARD DEVIATION
Error Clustering Ratio 1:1					
S1	8	1.00000000	1.00000000	1.00000000	0.00000000
S2	8	1.17690000	1.06000000	1.11073750	0.03578946
S3	8	1.00000000	1.00000000	1.00000000	0.00000000
S4	8	1.12700000	1.02980000	1.07657500	0.02992446
C1	8	1.21180000	1.08390000	1.16238750	0.04709478
C2	8	1.17130000	1.06850000	1.12958750	0.03841125
C3	8	1.16500000	1.05670000	1.11872500	0.04019153
C4	8	1.12820000	1.04570000	1.09213750	0.03146236
Error Clustering Ratio 2:1					
S1	8	1.00000000	1.00000000	1.00000000	0.00000000
S2	8	1.27330000	1.11580000	1.17772500	0.05948397
S3	8	1.00000000	1.00000000	1.00000000	0.00000000
S4	8	1.24300000	1.08690000	1.14695000	0.05740565
C1	8	1.20500000	1.08560000	1.14630000	0.04157032
C2	8	1.16650000	1.07120000	1.12372500	0.03268638
C3	8	1.15440000	1.05780000	1.11185000	0.03241481
C4	8	1.12120000	1.04750000	1.09247500	0.02603902
Error Clustering Ratio 1:2					
S1	8	1.00000000	1.00000000	1.00000000	0.00000000
S2	8	1.33480000	0.98820000	1.13238750	0.12800686
S3	8	1.00000000	1.00000000	1.00000000	0.00000000
S4	8	1.32600000	0.95990000	1.10585000	0.13809068
C1	8	1.22270000	1.03210000	1.12383750	0.07689103
C2	8	1.17190000	1.03160000	1.09645000	0.05631003
C3	8	1.16330000	1.03140000	1.08990000	0.05377859
C4	8	1.12180000	1.03070000	1.06875000	0.03731262

the 95 percent confidence level (C2) were tighter than their corresponding Stringer bound by the greatest amount. The 2:1 error clustering ratio provided the greatest amount of tightness for the modified Stringer bounds (S2,S4) and the modified DUS-cell bound at 85 percent nominal confidence (S4).

<u>Mean Taint Effects on Relative Tightness</u>. To determine the effects of mean taint on relative tightness, the study populations were sorted by mean taint. The relative tightness of each bound for each mean taint was then examined based on the maximum value, minimum value, mean and standard deviation of relative tightness. Table 3.12 shows these values of relative tightness suggesting that mean taint has little effect on the relative tightness of the bounds. For example, all four values of maximum, minimum, mean and standard deviation of relative tightness are extremely close to one another for all the bounds with different mean taints, indicating that the relative tightness is independent of the mean taint.

<u>Error Amount Intensity Effects on Relative Tightness</u>. To examine the possible effects of error amount intensity on relative tightness, the study populations were sorted by error amount intensity and the relative tightness of the bounds was examined. Table 3.13 lists the maximum value, minimum value, mean, and standard deviation of relative tightness for each bound within the three categories of

Table 3.12

Relative Tightness of Bounds by Mean Taint

BOUND	N	MAXIMUM VALUE	MINIMUM VALUE	MEAN	STANDARD DEVIATION
Mean Taint of .4/.2					
S1	12	1.00000000	1.00000000	1.00000000	0.00000000
S2	12	1.33480000	0.9882[illegible]000	1.12975833	0.08616874
S3	12	1.00000000	1.00000000	1.00000000	0.00000000
S4	12	1.32600000	0.95990000	1.09975000	0.08985617
C1	12	1.22270000	1.03210000	1.13660000	0.06294526
C2	12	1.17190000	1.03160000	1.10990000	0.04970487
C3	12	1.16500000	1.03140000	1.10073333	0.04836831
C4	12	1.12820000	1.03070000	1.07913333	0.03697174
Mean Taint of .2/.1					
S1	12	1.00000000	1.00000000	1.00000000	0.00000000
S2	12	1.32940000	1.06000000	1.15080833	0.08675888
S3	12	1.00000000	1.00000000	1.00000000	0.00000000
S4	12	1.31790000	1.02640000	1.11983333	0.09124440
C1	12	1.22000000	1.03820000	1.15175000	0.05226331
C2	12	1.16750000	1.03740000	1.12327500	0.03902976
C3	12	1.16030000	1.03710000	1.11291667	0.03802991
C4	12	1.12230000	1.03610000	1.08977500	0.02806716

Table 3.13

Relative Tightness of Bounds by Error Amount Intensity(EAI)

BOUND	N	MAXIMUM VALUE	MINIMUM VALUE	MEAN	STANDARD DEVIATION
EAI is Low					
S1	8	1.00000000	1.00000000	1.00000000	0.00000000
S2	8	1.33480000	1.13480000	1.23147500	0.07881200
S3	8	1.00000000	1.00000000	1.00000000	0.00000000
S4	8	1.32600000	1.08570000	1.19801250	0.09350935
C1	8	1.21130000	1.03210000	1.13508750	0.07057892
C2	8	1.17130000	1.03160000	1.11643750	0.05597744
C3	8	1.16500000	1.03140000	1.11177500	0.05256631
C4	8	1.12820000	1.03070000	1.09350000	0.03962095
EAI is Medium					
S1	8	1.00000000	1.00000000	1.00000000	0.00000000
S2	8	1.13230000	1.06000000	1.09356250	0.02665493
S3	8	1.00000000	1.00000000	1.00000000	0.00000000
S4	8	1.10040000	1.02640000	1.05700000	0.02867612
C1	8	1.22270000	1.15160000	1.19257500	0.02718859
C2	8	1.17190000	1.11830000	1.15001250	0.02106334
C3	8	1.16330000	1.10600000	1.13780000	0.02240057
C4	8	1.12180000	1.07950000	1.10436250	0.01640461
EAI is High					
S1	8	1.00000000	1.00000000	1.00000000	0.00000000
S2	8	1.15720000	0.98820000	1.09581250	0.05191794
S3	8	1.00000000	1.00000000	1.00000000	0.00000000
S4	8	1.13790000	0.95990000	1.07436250	0.05509913
C1	8	1.12980000	1.07560000	1.10486250	0.02137134
C2	8	1.10340000	1.05930000	1.08331250	0.01760848
C3	8	1.08830000	1.05080000	1.07090000	0.01455315
C4	8	1.06940000	1.03930000	1.05550000	0.01180968

error amount intensity Low error amount intensity had the greatest effect on the stability of the bounds. At this intensity level, that the greatest standard deviation and largest ranges of relative tightness were found for both the modified and unmodified DUS-cell bounds (C1,C2,C3,C4) and the modified Stringer bounds (S2,S4). However, at the medium error amount intensity level, both the modified and unmodified DUS-cell bounds (C1,C2,C3,C4) were the tightest relative to the corresponding Stringer bound. The modified Stringer bounds (S2,S4) provided the greatest amount of tightness relative to their corresponding unmodified Stringer bounds at the low error amount intensities, more than doubling the mean relative tightness of the modified Stringer bounds (S2,S4) at the medium and high error amount intensities.

To summarize the results of research question four, line item error rate had the most significant impact on the performance of the Stringer and DUS-cell bounds. Mean taint had little impact on coverage and relative tightness. The 1 percent error rate, 1:2 error clustering ratio, and low error amount intensity were associated with high coverages, yet at the same time, they were associated with great variability in relative tightness.

IV. Summary and Conclusions

This chapter consists of three major sections. The first section is a summary of the findings of the research. The second section contains the conclusions based on the analysis of the findings. The final section suggests possible research for the future.

Summary

The unmodified Stringer bounds at both levels of nominal confidence were the only robust bounds for the study populations examined. However, the unmodified DUS-cell bounds at both nominal confidence levels failed just once to provide adequate coverage. The modified Stringer bound failed once at the 95 percent nominal confidence and three times at the 85 percent nominal confidence. The modified DUS-cell bounds at both levels of confidence failed six times each. All coverage failures occurred in study populations with a line item error rate of 50 percent.

Both the modified and unmodified DUS-cell bounds were relatively tighter than the corresponding Stringer bounds for all study populations. However, the modified Stringer and modified DUS-cell bounds, at both levels of nominal confidence, were relatively looser than the unmodified Stringer and unmodified DUS-cell bounds, respectively, for study population 11. The level of nominal confidence

specified by the auditor had little effect on the unmodified DUS-cell bound. On the other hand, the modified DUS-cell bound at 85 percent nominal confidence attained an increase in average coverage twice that of the average coverage increase for the modified DUS-cell bound at 95 percent nominal confidence level.

Of the three error distribution characteristics the line item error rate of 50 percent caused the greatest decrease in coverage. Furthermore, the error clustering ratio of 2:1 also caused a significant decrease in coverage by the bounds. On the other hand, error taint had little effect on coverage.

As for relative tightness, the line item error rate again had the most significant impact on the mean relative tightness. In particular, a line item error rate of 1 percent had the greatest mean relative tightness for the modified Stringer bounds at both levels of nominal confidence, but the 30 percent line item error rate had the largest mean relative tightness for both the modified and unmodified DUS-cell bounds at both levels of nominal confidence. The error clustering ratio also had some effect on the mean relative tightness of the bounds. For instance, the error clustering ratio of 1:1 had a large mean relative tightness for the unmodified DUS-cell bounds at both nominal confidence levels and the modified DUS-cell bound at 95 percent nominal confidence.

The error clustering ratio of 2:1 had a large mean relative tightness for the modified Stringer bound at both nominal confidence levels and the modified DUS-cell bound at 85 percent nominal confidence. Error taint had little impact on the mean relative tightness just as it had little impact on coverage.

The population characteristic of error amount intensity had a significant impact on both coverage and mean relative tightness. High error amount intensity caused the greatest decrease in coverage as this is the intensity level which all insufficient coverages possessed. As for relative tightness, the low error amount intensity had the largest mean relative tightness for the modified Stringer bound at both nominal confidence levels. But it was the medium error amount intensity that had the greatest mean relatve tightness for both the modified and unmodified DUS-cell bounds at both nominal confidence levels.

Conclusions

Based on the findings, the bound that an auditor favors will depend on the individual's willingness to accept risk. An auditor unwilling to take chances would more than likely want to use the unmodified Stringer bound. In contrast, an auditor who is willing to accept a little more risk could reduce his upper error limit bound by 10 to 14 percent on the average by using the unmodified DUS-cell bound. That is, an auditor who could accept an overall mean coverage of

.9895 of the unmodified DUS-cell bound instead of the .9999 that the unmodified Stringer bound offers could achieve bounds 10 to 14 percent relatively tighter on the average, allowing the auditor to accept more actul book values as proper representations of the of the financial condition of the client. An auditor whose prior knowledge leads him to believe that the population he is examining is very unlikely to contain a 50 percent line item error rate may want to use the modified DUS-cell bound which was robust for all study populations with the exception of those with a 50 percent line item error rate. An auditor who is sure that the population does not have a 50 percent line item error rate could effectively use the modified DUS-cell bound and achieve bounds 20 to 28 percent tighter relative to the unmodified Stringer bound.

However, modified bounds are not recommended at this time for twc specific reasons. First, for one study population the modified Stringer and modified DUS-cell bounds were relatively looser than the unmodified Stringer bound and the unmodified DUS-cell bounds, respectively. Second, nominal confidence affected the performance of the modified bounds. The DUS-cell bound should be used instead of the Stringer bound even though it was not robust. It provided mean coverages of .9895 and .9582 for the 95 percent and 85 percent nominal confidence levels, respectively. It was also .1442 and .1068 percent tighter than the Stringer bound

for the 95 percent and 85 percent nominal confidence levels, respectively. The DUS-cell bound should also be used instead of the modified Stringer bound because the relative tightnesses of the two bounds were almost identical. Specifically, the DUS-cell bounds were .1442 and .1068 tighter than the unmodified Stringer bounds at the 95 percent and 85 percent nominal confidence levels, respectively. The modified Stringer bounds were .1403 and .1098 percent tighter than the unmodified Stringer bounds at the 95 percent and 85 percent nominal confidence levels, respectively. In addition to providing similar relative tightnessess, the unmodified DUS-cell bound, at both nominal confidence levels, failed only twice to provide adequate coverage, one at each level. The modified Stringer, at both confidence levels, failed to provide adequate coverage four times, once at the 95 percent nominal confidence level and three times at the 85 percent nominal confidence level. The DUS-cell bound should also be used instead of the modified DUS-cell bound because of the twelve insufficient coverages of the modified DUS-cell bound (six at each nominal confidence level).

Suggestions for Future Research.

The following are suggestions for future research:

A. How does the non-Bayesian DUS-cell bound perform relative to the Bayesian Cox and Snell model?

B. How does sample size affect the coverage and relative tightness of the DUS-cell bound relative to the Stringer bound?

Appendix A: Computer Program 1

```
c         AUTHOR:  MIKE HELTON
c
c
c         This program is designed to randomly select line
c         items to put error in for my thesis.
c
c     Key
c         mid = midpoint of $ value
c         cnt = # of line items in bvary
c         sum = total $ amount
c         bvary(9000) = array with book values
c         ranary(7000) = array with book values to be sampled
c         cntit = # of line items in low values
c         cntem = # of line items in high values
c         i,s,t,p are counters
c         ix,iy = values used in subroutine randu
c         yfl = # from subroutine between 0 and 1
c
c
c
c     Initialization of variables
         real mid,bvary,yfl
         real totary,errary,b,c
         integer i,s,p,ix,iy,t,cnt,cntit,cntem,ranary,index
         dimension bvary(9000),errary(9000),totary(9000)
         dimension ranary(7000),index(9000)
c
c
         mid = 0.0
         data i,s,p,t,cnt,cntit,cntem /7*0/
         ix = 12345
         yfl = 0.0
         iy = 0
c
c
c
      open(1,file='pop1.dat')
      rewind 1
      print *,'the file is open and rewound'
      open(2,file='sp04.dat')
      rewind 2
      i = 1
   10 read(1,72,end = 77) bv,error,tot,ind
   72 format(1x,f12.2,2x,f12.2,2x,f12.2,2x,i5)
      bvary(i) = bv
      errary(i) = 0.0
      totary(i) = tot
      index(i) = i
```

```
      i = i + 1
      cnt = cnt + 1
      go to 10
c
c
 77   print *,'the data has been read'
      mid = totary(cnt)/2
      print *,'the mid value is',mid
      i=1
 12   if (mid .ge. totary(i)) then
         cntit = cntit + 1
         i = i + 1
         go to 12
      else
         Print *, 'The mid value is',mid,'and it occurs at
     *line #',cntit
      end if
c
      cntem = cnt - cntit
      print *,'The cnt is',cnt,'and cntit is',cntit,'and
     *cntem is',cntem
      p = 1
      j = 1
      b = 1.0
      ix = 12345
      call randu (ix,iy,yfl)
 18   if (j .le. cntit .and. (b-1)/cntit .lt. .8) then
         if (index(j) .eq. 0) then
            j = j + 1
            if (j .gt. cntit) j=1
            go to 18
         end if
         ix = iy
         call randu (ix,iy,yfl)
         if (yfl .lt. .5) then
            ranary(p) = index(j)
            p = p + 1
            b = b + 1
            index(j) = 0
            j = j + 1
            if (j .gt. cntit) j=1
         else
            j = j + 1
            if (j .gt. cntit) j=1
         end if
      go to 18
      end if
c
c
      s = p - 1
      c = b
      t = 1 + cntit
```

```
19  if(t .ge. cntit .and. t .le. cnt .and. (c-p)/cntem
   *.lt. .8) then
       if (index(t) .eq. Ø) then
          t = t + 1
          if (t .gt. cnt) t = 1 + cntit
          go to 19
       end if
       ix = iy
       call randu(ix,iy,yfl)
       if (yfl .lt. .5) then
          s = s + 1
          ranary(s) = index(t)
          c = c + 1
          index(t) = Ø
          t = t + 1
          if (t .gt. cnt) t = 1 + cntit
       else
          t = t + 1
          if (t .gt. cnt) t = 1 + cntit
       end if
       go to 19
    end if
    do 25 m=1,s
       print *,ranary(m)
25  continue
    do 86 g=1,s
    write(2,15)ranary(g),cntit,cnt
15  format(1x,3(i6,2x))
86  continue
    print *,'S has a value of',s
    close (1)
    close (2)
    end

    subroutine randu(ix,iy,yfl)
    iy = ix*65539
    if (iy)5,6,6
 5  iy = iy + 2147483647 + 1
 6  yfl = iy
    yfl = yfl*.4656613E-9
    return
    end
```

Appendix B: Computer Program 2

```
C      PROGRAM STUDY
C***********************************************************
C
C     PURPOSE:
C
C     REVISION: V2.0
C
C     DATE CREATED: 26 MAR 84
C     DATE REVISIED: 26 APR 84
C
C     AUTHOR:  MAJ JEFF PHILLIPS
C
C     PROGRAM MODIFICATIONS: 1LT HAL STALCUP
C
C     REVISED:  MIKE HELTON
C
C***********************************************************
      COMMON A(350),B(350),TBV(20),M,NX,TY,JJ,LL,MM,ITT
      COMMON XLTLA(11),XLTLB(11),PI(11),TYE,TE
      COMMON XMU(11),CS(11),XMCS(11),F(200,11),FA(200,11)
      COMMON YTR2CS(11),YCS(11),YTR4CS(11),YMCS(11)
      COMMON BVARY(9000),ERRARY(9000),IYI(9000),IYM(9000)
      COMMON XTR1CS(11),XTR2CS(11),XTR3CS(11),XTR4CS(11)
      COMMON TELI,TEDV,TBR,BA(350),IRND(9000),BIGBV,BIGER
      COMMON IYJ(9000),IYK(9000),IYL(9000),INDEX(9000)
      COMMON TOTARY(9000),XMDPT(20),MIDL(20),IENDPT(20)
      COMMON PR(5),XMOO(2),AVARY(9000),TNTARY(9000),ISDYPP,
     *IENDD
      COMMON BBVARY(500),AAVARY(500),EERARY(50C)
      COMMON TTNTRY(500),TTOTRY(500),IINDEX(500),SKIP,
     *SIMPER,TYSKP
C***********************************************************
C  CALL TO SUBROUTINES BEGINS HERE:
C    INITA PUTS INITAL VALUES INTO CONSTANT ARRAYS.
C    ERRATE PUTS ERRORS INTO STUDY POPULATIONS.
C***********************************************************
      open(1,file='belch.hat')
      open(4,file='belch.dat')
      open(9,file='belch.fat')
      rewind 1
      rewind 4
      rewind 9
      write(9,100)
 100  format(//10x,'Begin simulation.........')
      CALL INITA
      CALL ERRATE
      STOP
      END
```

```
C
C
      SUBROUTINE SAMPL
C*************************************************************
C
C     SAMPL IS USED TO ACCESS THE POPULATIONS AND PRODUCE A
C     RANDOM SAMPLE FROM THE POPULATIONS.  SAMPL ALSO
C     CALCULATES THE BOUNDS USED BY THE NEXT SUBROUTINE.
C
C*************************************************************
      COMMON A(350),B(350),TBV(20),M,NX,TY,JJ,LL,MM,ITT
      COMMON XLTLA(11),XLTLB(11),PI(11),TYE,TE
      COMMON XMU(11),CS(11),XMCS(11),F(200,11),FA(200,11)
      COMMON YTR2CS(11),YCS(11),YTR4CS(11),YMCS(11)
      COMMON BVARY(9000),ERRARY(9000),IYI(9000),IYM(9000)
      COMMON XTR1CS(11),XTR2CS(11),XTR3CS(11),XTR4CS(11)
      COMMON TELI,TEDV,TBR,BA(350),IRND(9000),BIGBV,BIGER
      COMMON IYJ(9000),IYK(9000),IYL(9000),INDEX(9000)
      COMMON TOTARY(9000),XMDPT(20),MIDL(20),IENDPT(20)
      COMMON *PR(5),XMOO(2),AVARY(9000),TNTARY(9000),ISDYPP,
     *IENDD
      COMMON BBVARY(500),AAVARY(500),EERARY(500)
      COMMON TTNTRY(500),TTOTRY(500),IINDEX(500),SKIP,
     *SIMPER,TYSKP
      COMMON/STUDY1/ ZK, SAMPER, SAMPBV, XMLE, EFF, EFFA,
     *BP, BPA
      COMMON/STUDY2/ PGW, PGWA, IM, TOTANT, ITOOT, IGY, J,
     *L, N
      COMMON/STUDY3/ IX, IY, YFL, NS, FN
      COMMON/STUDY4/ JZZ, STR, XMSTR, STRA, XMSTRA
      COMMON UEL, UELA, MUEL, MUELA
C*************************************************************
      IX=54321
      CALL RANDU(IX,IY,YFL)
      BP=3.00
      BPA=1.90
      SAMPBV=BIGBV
      SAMPER=BIGER
      XMLE=0.0
      PGW=0.0
      PGWA=0.0
      EFF=0.0
      EFFA=0.0
      IM=0
      TOTANT=0.0
      ITOOT=0
      IGY=0
      J=0
      DO 4 JZZ=1,500
      DO 57 ITO=1,350
       A(ITO)=0.00
 57   CONTINUE
```

```
C
      CALL STEP1
C
      IGY=0
      DO 7 IK=1,N
      SAMPER=SAMPER+EERARY(IK)
      SAMPBV=SAMPBV+BBVARY(IK)
      IF (TTNTRY(IK).NE.0.0) THEN
         ITOOT=ITOOT+1
         A(ITOOT)=TTNTRY(IK)
       ELSE
          GO TO 7
      END IF
 7    CONTINUE
      IF (ITOOT.LE.1)  GO TO 33
C***********************************************************
C     SORT EACH SAMPLE IN DESCENDING ORDER BY TAINT
C***********************************************************
      CALL SORTA(A,350,ITOOT)
C***********************************************************
C     CALCULATE THE STRINGER(STR) AND MODIFIED
C     STRINGER(MSTR) BOUNDS
C***********************************************************
 33   CONTINUE
C
      CALL BOUND1
C
C***********************************************************
C     CALCULATE THE CELL AND MODIFIED CELL BOUNDS
C***********************************************************
      CALL BOUND2
C***********************************************************
C                   OUTPUT RESULTS
C***********************************************************
      ZNEG=TY-SAMPBV
      WRITE(4,15) JJ,ISDYPP,JZZ,LL,MM,ITT,N,TE
      WRITE(4,17) IM,TY,SAMPBV,SAMPER,SIMPER,ZNEG
 17   FORMAT(1X,I5,2X,5(F12.2,2X))
 15   FORMAT(1X,7(I4,2X),F12.2,2X)
      WRITE(4,16) STR,XMSTR,STRA,XMSTRA
      WRITE(4,16) UEL,MUEL,UELA,MUELA
 16   FORMAT(1X,5(F14.2,2X))
C
C
      SAMPER=BIGER
      SAMPBV=BIGBV
      XMLE=0.0
      PGW=0.0
      PGWA=0.0
      IM=0
      ITOOT=0
      TOTANT=0.0
```

```
 4    CONTINUE
      RETURN
      END
C
C
      SUBROUTINE BOUND2
C************************************************************
C
C
C************************************************************
      COMMON A(350),B(350),TBV(20),M,NX,TY,JJ,LL,MM,ITT
      COMMON XLTLA(11),XLTLB(11),PI(11),TYE,TE
      COMMON XMU(11),CS(11),XMCS(11),F(200,11),FA(200,11)
      COMMON YTR2CS(11),YCS(11),YTR4CS(11),YMCS(11)
      COMMON BVARY(9000),ERRARY(9000),IYI(9000),IYM(9000)
      COMMON XTR1CS(11),XTR2CS(11),XTR3CS(11),XTR4CS(11)
      COMMON TELI,TEDV,TBR,BA(350),IRND(9000),BIGBV,BIGER
      COMMON IYJ(9000),IYK(9000),IYL(9000),INDEX(9000)
      COMMON TOTARY(9000),XMDPT(20),MIDL(20),IENDPT(20)
      COMMON *PR(5),XMOO(2),AVARY(9000),TNTARY(9000),ISDYPP,
     *IENDD
      COMMON BBVARY(500),AAVARY(500),EERARY(500)
      COMMON TTNTRY(500),TTOTRY(500),IINDEX(500),SKIP,
     *SIMPER,TYSKP
      COMMON/STUDY1/ ZK, SAMPER, SAMPBV, XMLE, EFF, EFFA,
     *BP, BPA
      COMMON/STUDY2/ PGW, PGWA, IM, TOTANT, ITOOT, IGY, J,
     *L, N
      COMMON/STUDY3/ IX, IY, YFL, NS, FN
      COMMON/STUDY4/ JZZ, STR, XMSTR, STRA, XMSTRA
      COMMON UEL, UELA, MUEL, MUELA
C************************************************************
      real sumdt,las,lasa,dto,dtoa,ssv,ssva,suel,suela,fn
      real cumavg,dt,muela,muel
      integer it,iw,im
      sumdt=0.0
      las=0.0
      lasa=0.0
      dto=1.0
      dtoa=1.0
      ssv=3.00
      ssva=1.90
      suel=3.00
      suela=1.90
      fn=0.0
      muela=0.0
      muel=0.0
      sumb=0.0
      sumba=0.0
      im=0
      do 12 it = 1,n
      if (a(it) .gt. 0.00) im = im + 1
```

```
 12   CONTINUE
c
c
      do 13 iw = 1,im
      sumb = sumb + b(iw)
      sumba = sumba + ba(iw)
      uelf = bp + iw + sumb
      uelfa = bpa + iw + sumba
      dt = a(iw)
      sumdt = sumdt + a(iw)
      cumavg = sumdt/iw
      if (ssv .gt. las) then
      uelp = ssv
      else
      uelp = las
      end if
      if(ssva .gt. lasa) then
      uelpa = ssva
      else
      uelpa = lasa
      end if
      las = uelp + a(iw)
      lasa = uelpa + a(iw)
      ssv = uelf*cumavg
      ssva = uelfa*cumavg
      if(ssv .gt. las) then
      suel = ssv
      else
      suel = las
      end if
      if (ssva .gt. lasa) then
      suela = ssva
      else
      suela = lasa
      end if
  13  continue
      fn = float(n)
      uel = suel*tyskp/fn + simper
      uela = suela*tyskp/fn + simper
      muel = (ty-sampbv)*suel/fn + samper
      muela = (ty-sampbv)*suela/fn + samper
      return
      end
C
C
      SUBROUTINE BOUND1
C***********************************************************
C
C
C***********************************************************
      COMMON A(350),B(350),TBV(20),M,NX,TY,JJ,LL,MM,ITT
      COMMON XLTLA(11),XLTLB(11),PI(11),TYE,TE
```

```
      COMMON XMU(11),CS(11),XMCS(11),F(200,11),FA(200,11)
      COMMON YTR2CS(11),YCS(11),YTR4CS(11),YMCS(11)
      COMMON BVARY(9000),ERRARY(9000),IYI(9000),IYM(9000)
      COMMON XTR1CS(11),XTR2CS(11),XTR3CS(11),XTR4CS(11)
      COMMON TELI,TEDV,TBR,BA(350),IRND(9000),BIGBV,BIGER
      COMMON IYJ(9000),IYK(9000),IYL(9000),INDEX(9000)
      COMMON TOTARY(9000),XMDPT(20),MIDL(20),IENDPT(20)
      COMMON *PR(5),XMOO(2),AVARY(9000),TNTARY(9000),ISDYPP,
     *IENDD
      COMMON BBVARY(500),AAVARY(500),EERARY(500)
      COMMON TTNTRY(500),TTOTRY(500),IINDEX(500),SKIP,
     *SIMPER,TYSKP
      COMMON/STUDY1/ ZK, SAMPER, SAMPBV, XMLE, EFF, EFFA,
     *BP, BPA
      COMMON/STUDY2/ PGW, PGWA, IM, TOTANT, ITOOT, IGY, J,
     *L, N
      COMMON/STUDY3/ IX, IY, YFL, NS, FN
      COMMON/STUDY4/ JZZ, STR, XMSTR, STRA, XMSTRA
C***************************************************************
      DO 10 IZ=1,N
      XMLE=XMLE+A(IZ)
      PGW=PGW+(B(IZ)*A(IZ))
      PGWA=PGWA+(BA(IZ)*A(IZ))
 10   CONTINUE
      FN = FLOAT(N)
      STR=(BP+XMLE+PGW)*TYSKP/FN+SIMPER
      STRA=(BPA+XMLE+PGWA)*TYSKP/FN+SIMPER
      XMSTR=(TY-SAMPBV)*(BP+XMLE+PGW)/FN+SAMPER
      XMSTRA=(TY-SAMPBV)*(BPA+XMLE+PGWA)/FN+SAMPER
      RETURN
      END
C
C
      SUBROUTINE STEP1
C***************************************************************
C
C
C***************************************************************
      COMMON A(350),B(350),TBV(20),M,NX,TY,JJ,LL,MM,ITT
      COMMON XLTLA(11),XLTLB(11),PI(11),TYE,TE
      COMMON XMU(11),CS(11),XMCS(11),F(200,11),FA(200,11)
      COMMON YTR2CS(11),YCS(11),YTR4CS(11),YMCS(11)
      COMMON BVARY(9000),ERRARY(9000),IYI(9000),IYM(9000)
      COMMON XTR1CS(11),XTR2CS(11),XTR3CS(11),XTR4CS(11)
      COMMON TELI,TEDV,TBR,BA(350),IRND(9000),BIGBV,BIGER
      COMMON IYJ(9000),IYK(9000),IYL(9000),INDEX(9000)
      COMMON TOTARY(9000),XMDPT(20),MIDL(20),IENDPT(20)
      COMMON PR(5),XMOO(2),AVARY(9000),TNTARY(9000),ISDYPP,
     *IENDD
      COMMON BBVARY(500),AAVARY(500),EERARY(500)
      COMMON TTNTRY(500),TTOTRY(500),IINDEX(500),SKIP,
     *SIMPER,TYSKP
```

```
      COMMON/STUDY1/ ZK, SAMPER, SAMPBV, XMLE, EFF, EFFA,
     *BP, BPA
      COMMON/STUDY2/ PGW, PGWA, IM, TOTANT, ITOOT, IGY, J,
     *L, N
      COMMON/STUDY3/ IX, IY, YFL, NS, FN
      COMMON/STUDY4/ JZZ, STR, XMSTR, STRA, XMSTRA
C*************************************************************
 14   IX=IY
      CALL RANDU(IX,IY,YFL)
      IF (YFL.EQ.0.0) GO TO 14
      Y=YFL*SKIP
      DO 6 IMY=1,N
        TIJ=Y+(SKIP*(FLOAT(IMY)-1.0))
        CALL SEARCH(TIJ,LINEI)
        IGY=IGY+1
        BBVARY(IGY)=BVARY(LINEI)
        AAVARY(IGY)=AVARY(LINEI)
        EERARY(IGY)=ERRARY(LINEI)
        TTNTRY(IGY)=TNTARY(LINEI)
        TTOTRY(IGY)=TOTARY(LINEI)
        IINDEX(IGY)=INDEX(LINEI)
 6    CONTINUE
C     WRITE(1,100) JZZ,IGY
C     DO 60 IGH=1,IGY
C     WRITE(1,200) BBVARY(IGH),AAVARY(IGH),EERARY(IGH),
C    *TTNTRY(IGH),TTOTRY(IGH),IINDEX(IGH)
C60   CONTINUE
C100  FORMAT(10X,'INTERATION=',I5,'    SAMPLE SIZE=',I5)
C200  FORMAT(5X,6F12.2)
      RETURN
      END
C
      SUBROUTINE SEARCH(TIJ,LINEI)
C*************************************************************
C
C
C*************************************************************
      COMMON A(350),B(350),TBV(20),M,NX,TY,JJ,LL,MM,ITT
      COMMON XLTLA(11),XLTLB(11),PI(11),TYE,TE
      COMMON XMU(11),CS(11),XMCS(11),F(200,11),FA(200,11)
      COMMON YTR2CS(11),YCS(11),YTR4CS(11),YMCS(11)
      COMMON BVARY(9000),ERRARY(9000),IYI(9000),IYM(9000)
      COMMON XTR1CS(11),XTR2CS(11),XTR3CS(11),XTR4CS(11)
      COMMON TELI,TEDV,TBR,BA(350),IRND(9000),BIGBV,BIGER
      COMMON IYJ(9000),IYK(9000),IYL(9000),INDEX(9000)
      COMMON TOTARY(9000),XMDPT(20),MIDL(20),IENDPT(20)
      COMMON PR(5),XMOO(2),AVARY(9000),TNTARY(9000),ISDYPP,
     *IENDD
      COMMON BBVARY(500),AAVARY(500),EERARY(500)
      COMMON TTNTRY(500),TTOTRY(500),IINDEX(500),SKIP,
     *SIMPER,TYSKP
      COMMON/STUDY1/ ZK, SAMPER, SAMPBV, XMLE, EFF, EFFA,
```

```
     *BP, BPA
      COMMON/STUDY2/ PGW, PGWA, IM, TOTANT, ITOOT, IGY, J,
     *L, N
      COMMON/STUDY3/ IX, IY, YFL, NS, FN
      COMMON/STUDY4/ JZZ, STR, XMSTR, STRA, XMSTRA
C***************************************************************
      INTEGER MAXPT, MINPT, MIDPT
C
      MINPT = 1
      MAXPT = IENDD
   10 CONTINUE
      MIDPT = (MAXPT-MINPT)/2 + MINPT
      IF ( (MAXPT-MINPT).EQ.1) THEN
        LINEI = MAXPT
        RETURN
      ELSE
        IF (TIJ.LE.TOTARY(MIDPT)) THEN
          MAXPT = MIDPT
        ELSE
          MINPT = MIDPT
        ENDIF
      ENDIF
      GO TO 10
      END
C
C
      SUBROUTINE ERRATE
C
C
      COMMON A(350),B(350),TBV(20),M,NX,TY,JJ,LL,MM,ITT
      COMMON XLTLA(11),XLTLB(11),PI(11),TYE,TE
      COMMON XMU(11),CS(11),XMCS(11),F(200,11),FA(200,11)
      COMMON YTR2CS(11),YCS(11),YTR4CS(11),YMCS(11)
      COMMON BVARY(9000),ERRARY(9000),IYI(9000),IYM(9000)
      COMMON XTR1CS(11),XTR2CS(11),XTR3CS(11),XTR4CS(11)
      COMMON TELI,TEDV,TBR,BA(350),IRND(9000),BIGBV,BIGER
      COMMON IYJ(9000),IYK(9000),IYL(9000),INDEX(9000)
      COMMON TOTARY(9000),XMDPT(20),MIDL(20),IENDPT(20)
      COMMON PR(5),XMOO(2),AVARY(9000),TNTARY(9000),ISDYPP,
     *IENDD
      COMMON BBVARY(500),AAVARY(500),EERARY(500)
      COMMON TTNTRY(500),TTOTRY(500),IINDEX(500),SKIP,
     *SIMPER,TYSKP
C
C     JJ IS POPULATIONS 1 THRU 4 (ORDERED POPULATIONS)
C     LL IS ERROR RATES BY LINE ITEM (.50,.30,.15,.01)
C     MM IS ERROR RATE DISTRIBUTION (UNIFORM,DECREASING,
C     INCREASING)
C     ITT IS TAINTINGS (.40,.20 AND .20,.10)
C     KK IS STRATA (STRATA 1=LOW BV'S;STRATA 2=HIGH BV'S)
C
      integer hasit
```

```
      dimension hasit(9000)
      J=0
      I=0
      KNT=0
      KUZ=0
      KAT=0
      MID=0
      IENDD=0
      EEXP=0.0
      ISDYPP=0
      IX=12345
      CALL RANDU(IX,IY,YFL)
      DO 532 JJ=1,1
      IF (JJ.EQ.1) then
        OPEN(7,FILE='pop1.dat')
        write(9,201)
201     format(/1x,'Opened pop1, processing..........')
      endif
      IF (JJ.EQ.2) then
        OPEN(7,FILE='pop2.dat')
        write(9,202)
202     format(/1x,'Opened pop2, processing..........')
      endif
      IF (JJ.EQ.3) then
        OPEN(7,FILE='pop3.dat')
        write(9,203)
203     format(/1x,'Opened pop3, processing..........')
      endif
      IF (JJ.EQ.4) then
        OPEN(7,FILE='pop4.dat')
        write(9,204)
204     format(/1x,'Opened pop4, processing..........')
      endif
      REWIND 7
54    READ(7,72,END=55) BV,ERROR,TOT,IND
72    FORMAT(1X,F12.2,2X,F12.2,2X,F12.2,2X,I5)
      I=I+1
      BVARY(I)=BV
      ERRARY(I)=0.0
      TOTARY(I)=TOT
      AVARY(I)=0.0
      TNTARY(I)=0.0
      INDEX(I)=I
      IRND(I)=0
      GO TO 54
55    CONTINUE
      CLOSE(7)
      write(9,200) i
200   format(/1x,'Processed ',i5,' records.')
      IF (JJ.EQ.1) then
        OPEN(8,FILE='sp04.dat')
        write(9,301)
```

```
301     format(/1x,'Opened sp04, processing..........')
      endif
      IF (JJ.EQ.2) then
        OPEN(8,FILE='sp05.dat')
        write(9,302)
302     format(/1x,'Opened sp05, processing..........')
      endif
      IF (JJ.EQ.3) then
        OPEN(8,FILE='sp06.dat')
        write(9,303)
303     format(/1x,'Opened sp06, processing..........')
      endif
      IF (JJ.EQ.4) then
        OPEN(8,FILE='sp02.dat')
        write(9,304)
304     format(/1x,'Opened sp02, processing..........')
      endif
      REWIND 8
      I=0
999   READ(8,73,END=64) IIW,MID,IENDD
73    FORMAT(1X,3(I6,2X))
      i = i + 1
      IRND(IIW)=IIW
      GO TO 999
64    CONTINUE
      write(9,300) i
300   format(/1x,'Processed ',i5,' records.')
      i = 0
      CLOSE(8)
      DO 533 LL=1,4
      IF (JJ.EQ.1) THEN
      IF (LL.EQ.1) THEN
         PR(1)=.50684
         PR(2)=.25342
         PR(3)=.50
         PR(4)=.486855
         PR(5)=.973710
      END IF
      IF (LL.EQ.2) THEN
         PR(1)=.304105
         PR(2)=.152053
         PR(3)=.30
         PR(4)=.292113
         PR(5)=.584226
       END IF
      IF (LL.EQ.3) THEN
         PR(1)=.152053
         PR(2)=.076026
         PR(3)=.15
         PR(4)=.146056
         PR(5)=.292113
      END IF
```

```
IF (LL.EQ.4) THEN
   PR(1)=.010137
   PR(2)=.005068
   PR(3)=.01
   PR(4)=.009737
   PR(5)=.019474
END IF
END IF
IF (JJ.EQ.2) THEN
   IF (LL.EQ.1) THEN
      PR(1)=.520888
      PR(2)=.260444
      PR(3)=.50
      PR(4)=.462877
      PR(5)=.925754
   END IF
   IF (LL.EQ.2) THEN
      PR(1)=.312532
      PR(2)=.156266
      PR(3)=.30
      PR(4)=.277726
      PR(5)=.555452
   END IF

   IF (LL.EQ.3) THEN
      PR(1)=.156266
      PR(2)=.078133
      PR(3)=.15
      PR(4)=.138863
      PR(5)=.277726
   END IF
   IF (LL.EQ.4) THEN
      PR(1)=.010418
      PR(2)=.005209
      PR(3)=.01
      PR(4)=.009258
      PR(5)=.018515
   END IF
END IF
IF (JJ.EQ.3) THEN
  IF (LL.EQ.1) THEN
     PR(1)=.512059
     PR(2)=.256029
     PR(3)=.50
     PR(4)=.477509
     PR(5)=.955019
  END IF
  IF (LL.EQ.2) THEN
     PR(1)=.307235
     PR(2)=.153618
     PR(3)=.30
     PR(4)=.286506
```

```
      PR(5)=.573011
    END IF
    IF (LL.EQ.3) THEN
      PR(1)=.153618
      PR(2)=.076809
      PR(3)=.15
      PR(4)=.143253
      PR(5)=.286506
    END IF
    IF (LL.EQ.4) THEN
      PR(1)=.010241
      PR(2)=.005121
      PR(3)=.01
      PR(4)=.009550
      PR(5)=.019100
    END IF
  END IF
  IF (JJ.EQ.4) THEN
    IF (LL.EQ.1) THEN
      PR(1)=.508408
      PR(2)=.254204
      PR(3)=.50
      PR(4)=.483992
      PR(5)=.967984
    END IF
    IF (LL.EQ.2) THEN
      PR(1)=.305045
      PR(2)=.152522
      PR(3)=.30
      PR(4)=.290395
      PR(5)=.580790
    END IF
    IF (LL.EQ.3) THEN
      PR(1)=.152522
      PR(2)=.076261
      PR(3)=.15
      PR(4)=.145197
      PR(5)=.290395
    END IF
    IF (LL.EQ.4) THEN
      PR(1)=.010168
      PR(2)=.005084
      PR(3)=.01
      PR(4)=.009679
      PR(5)=.019359
    END IF
  END IF
  DO 534 MM=1,3
  IF (MM.EQ.1) THEN
    P=PR(3)
    PP=PR(3)
  END IF
```

```
      IF (MM.EQ.2) THEN
         P=PR(1)
         PP=PR(2)
      END IF
      IF (MM.EQ.3) THEN
         P=PR(4)
         PP=PR(5)
      END IF
      DO 405 ITT=1,2
      IF (ITT.EQ.1) THEN
         XMOO(1)=.8131
         XMOO(2)=.2083
      ELSE
         XMOO(1)=.2083
         XMOO(2)=.10
      END IF
      KNT=0
      KUZ=0
      KAT=0
      do 8 r = 1,9000
      hasit(r)=0
 8    continue
      DO 502 KK=1,IENDD
      ix = iy
      call randu(ix,iy,yfl)
      if (yfl .lt. .5) go to 502
      if (hasit(kk) .gt. 0) go to 502
      IZZ=INT(P*0.1)
      IF (KK.LE.MID) THEN
         IJKLM=INT(FLOAT(MID)*P)
      ELSE
         IJKLM=INT(FLOAT(IENDD-MID)*PP)
      END IF
      IF (KK.LE.MID) THEN
         IF ((KNT.LE.IJKLM).AND.(IRND(KK).NE.0)) THEN
            IF (KUZ.LE.IZZ) THEN
               AVARY(KK)=BVARY KK)+(BVARY(KK)*1.0)
               KUZ=KUZ+1
               KNT=KNT+1
            ELSE
1000           IX=IY
               CALL RANDU(IX,IY,YFL)
               IF (YFL.EQ.0.0) GO TO 1000
               EEXP=-XMOO(1)*ALOG(YFL)
               IF (EEXP.GE.1.0) GO TO 1000
               AVARY(KK)=BVARY(KK)+(BVARY(KK)*EEXP)
               KNT=KNT+1
            END IF
          ELSE
          AVARY(KK)=BVARY(KK)
          END IF
        END IF
```

```
          IF (KK.GT.MID) THEN
             IF ((KAT.LE.IJKLM).AND.(IRND(KK).NE.0)) THEN
 2000        IX=IY
             CALL RANDU(IX,IY,YFL)
             IF (YFL.EQ.0.0) GO TO 2000
             EEXP=-XMOO(2)*ALOG(YFL)
             IF (EEXP.GE.1.0) GO TO 2000
             AVARY(KK)=BVARY(KK)+(BVARY(KK)*EEXP)
             KAT=KAT+1
          ELSE
             AVARY(KK)=BVARY(KK)
           END IF
          END IF
         ERRARY(KK)=AVARY(KK)-BVARY(KK)
         TNTARY(KK)=ERRARY(KK)/BVARY(KK)
         if(kk .ge. mid .and. knt .lt. int(float(mid)*p)) kk=1
         if(kk .eq. iendd .and. kat .lt. ijklm) kk = mid + 1
         hasit(kk) = kk
  502    CONTINUE
         ISDYPP=ISDYPP+1
         CALL INFO
         CALL STRPOF
         write(9,600)
  600    format(1x,'Processing sample, please standby.......')
         CALL SAMPL
  405    CONTINUE
  534    CONTINUE
  533    CONTINUE
         close(4)
  532    CONTINUE
         RETURN
         END
C
C
C
         SUBROUTINE RANDU(IX,IY,YFL)
         IY=IX*65539
         IF (IY)5,6,6
    5    IY=IY+2147483647+1
    6    YFL=IY
         YFL=YFL*.4656613E-9
         RETURN
         END
C
C
C
         SUBROUTINE SORTA(A,ND,NS)
C
         REAL A(ND), TEMP
         INTEGER I, LASTS, LASTI, SSTART
         LOGICAL INSORT
C
```

```
      SSTART = NS - 1
      LASTS = 1
      LASTI = LASTS
      INSORT = .FALSE.
 10   CONTINUE
      IF (.NOT.INSORT) THEN
        INSORT = .TRUE.
        DO 20 I = SSTART, LASTI, -1
        IF (A(I).LT.A(I+1)) THEN
          TEMP = A(I)
          A(I) = A(I+1)
          A(I+1) = TEMP
          INSORT = .FALSE.
          LASTS = I
        ENDIF
 20     CONTINUE
        LASTI = LASTS+1
        GO TO 10
      ENDIF
      RETURN
      END
C
C
C
      SUBROUTINE INFO
      COMMON A(350),B(350),TBV(20),M,NX,TY,JJ,LL,MM,ITT
      COMMON XLTLA(11),XLTLB(11),PI(11),TYE,TE
      COMMON XMU(11),CS(11),XMCS(11),F(200,11),FA(200,11)
      COMMON YTR2CS(11),YCS(11),YTR4CS(11),YMCS(11)
      COMMON BVARY(9000),ERRARY(9000),IYI(9000),IYM(9000)
      COMMON XTR1CS(11),XTR2CS(11),XTR3CS(11),XTR4CS(11)
      COMMON TELI,TEDV,TBR,BA(350),IRND(9000),BIGBV,BIGER
      COMMON IYJ(9000),IYK(9000),IYL(9000),INDEX(9000)
      COMMON TOTARY(9000),XMDPT(20),MIDL(20),IENDPT(20)
      COMMON PR(5),XMOO(2),AVARY(9000),TNTARY(9000),ISDYPP,
     *IENDD
      COMMON BBVARY(500),AAVARY(500),EERARY(500)
      COMMON TTNTRY(500),TTOTRY(500),IINDEX(500),SKIP,
     *SIMPER,TYSKP
C     M IS # OF ERRORS
C     NX IS # OF LINE ITEMS
C     TY IS STUDY POPULATION TOTAL BOOK VALUE
C     TYE IS STUDY POPULATION TOTAL BOOK VALUE IN ERROR
C     TE IS TOTAL ERROR VALUE IN STUDY POPULATION
      NX=0
      TE=0.0
      TY=0.0
      M=0
      TYE=0.0
      TELI=0.0
      TEDV=0.0
      TBR=0.0
```

```
      TETYY=0.0
      DO 1 I=1,IENDD
      NX=NX+1
      TE=TE+ERRARY(I)
      TY=TY+BVARY(I)
      IF (ERRARY(I).NE.0.0) THEN
         M=M+1
         TYE=TYE+BVARY(I)
      END IF
 1    CONTINUE
      TELI=FLOAT(M)/FLOAT(NX)
      TEDV=TYE/TY
      TBR=TE/TYE
      TETYY=TE/TY
      write(9,19) JJ,LL,MM,ITT
      WRITE(9,20) ISDYPP,M,NX,TY,TYE,XMOO(2)
      WRITE(9,21) TE,TELI,TEDV,TBR,TETYY
 19   FORMAT(1X,I2,2X,I2,2X,I2,2X,I2)
 20   FORMAT(1X,3(I5,2X),3(F14.4,2X))
 21   FORMAT(1X,F14.2,4(F14.6,1X))
      RETURN
      END
C
C
C
      SUBROUTINE INITA
C*************************************************************
C
C     INITA IS DESIGNED TO LOAD THE CONSTANT ARRAYS BELOW
C     WITH THE VALUES NECESSARY FOR THE SIMULATION EFFORT.
C
C     THE ARRAYS LOADED HERE ARE NOT CHANGED IN THE PROGRAM
C     WHICH MEANS THAT INITA NEED ONLY BE CALLED ONCE.
C
C*************************************************************
      COMMON A(350),B(350),TBV(20),M,NX,TY,JJ,LL,MM,ITT
      COMMON XLTLA(11),XLTLB(11),PI(11),TYE,TE
      COMMON XMU(11),CS(11),XMCS(11),F(200,11),FA(200,11)
      COMMON YTR2CS(11),YCS(11),YTR4CS(11),YMCS(11)
      COMMON BVARY(9000),ERRARY(9000),IYI(9000),IYM(9000)
      COMMON XTR1CS(11),XTR2CS(11),XTR3CS(11),XTR4CS(11)
      COMMON TELI,TEDV,TBR,BA(350),IRND(9000),BIGBV,BIGER
      COMMON IYJ(9000),IYK(9000),IYL(9000),INDEX(9000)
      COMMON TOTARY(9000),XMDPT(20),MIDL(20),IENDPT(20)
      COMMON PR(5),XMOO(2),AVARY(9000),TNTARY(9000),ISDYPP,
     *IENDD
      COMMON BBVARY(500),AAVARY(500),EERARY(500)
      COMMON TTNTRY(500),TTOTRY(500),IINDEX(500),SKIP,
     *SIMPER,TYSKP
C*************************************************************
      B(1)=.75
      B(2)=.55
```

```
B(3)=.46
B(4)=.40
B(5)=.36
B(6)=.33
B(7)=.30
B(8)=.29
B(9)=.27
B(10)=.26
B(11)=.24
B(12)=.24
B(13)=.22
B(14)=.22
B(15)=.21
B(16)=.21
B(17)=.19
B(18)=.20
B(19)=.18
B(20)=.19
B(21)=.18
B(22)=.17
B(23)=.17
B(24)=.17
B(25)=.16
B(26)=.16
B(27)=.16
B(28)=.15
B(29)=.16
B(30)=.15
B(31)=.15
B(32)=.15
B(33)=.15
B(34)=.15
B(35)=.15
B(36)=.15
B(37)=.15
B(38)=.15
B(39)=.15
B(40)=.13
B(41)=.13
B(42)=.13
B(43)=.13
B(44)=.13
B(45)=.12
B(46)=.12
B(47)=.12
B(48)=.12
B(49)=.12
B(50)=.12
B(51)=.12
B(52)=.12
B(53)=.12
B(54)=.12
```

B(55)=.12
B(56)=.12
B(57)=.12
B(58)=.12
B(59)=.12
B(60)=.11
B(61)=.11
B(62)=.11
B(63)=.11
B(64)=.11
B(65)=.10
B(66)=.10
B(67)=.10
B(68)=.10
B(69)=.10
B(70)=.10
B(71)=.10
B(72)=.10
B(73)=.10
B(74)=.10
B(75)=.10
B(76)=.10
B(77)=.10
B(78)=.10
B(79)=.10
B(80)=.09
B(81)=.09
B(82)=.09
B(83)=.09
B(84)=.09
B(85)=.09
B(86)=.09
B(87)=.09
B(88)=.09
B(89)=.09
B(90)=.09
B(91)=.09
B(92)=.09
B(93)=.09
B(94)=.09
B(95)=.08
B(96)=.08
B(97)=.08
B(98)=.08
B(99)=.08
B(100)=.08
B(101)=.08
B(102)=.08
B(103)=.08
B(104)=.08
B(105)=.08
B(106)=.07

```
B(1Ø7)=.Ø7
B(1Ø8)=.Ø7
B(1Ø9)=.Ø7
B(11Ø)=.Ø7
B(111)=.Ø7
B(112)=.Ø7
B(113)=.Ø7
B(114)=.Ø7
B(115)=.Ø7
  BA(1)=.48
  BA(2)=.35
  BA(3)=.29
  BA(4)=.25
  BA(5)=.23
  BA(6)=.21
  BA(7)=.19
  BA(8)=.18
  BA(9)=.17
  BA(1Ø)=.17
  BA(11)=.15
  BA(12)=.15
  BA(13)=.14
  BA(14)=.14
  BA(15)=.13
  BA(16)=.13
  BA(17)=.13
  BA(18)=.12
  BA(19)=.12
  BA(2Ø)=.11
  BA(21)=.11
  BA(22)=.11
  BA(23)=.11
  BA(24)=.11
  BA(25)=.1Ø
  BA(26)=.1Ø
  BA(27)=.1Ø
  BA(28)=.1Ø
  BA(29)=.1Ø
  BA(3Ø)=.Ø9
  BA(31)=.Ø9
  BA(32)=.Ø9
  BA(33)=.Ø9
  BA(34)=.Ø9
  BA(35)=.Ø9
  BA(36)=.Ø9
  BA(37)=.Ø9
  BA(38)=.Ø9
  BA(39)=.Ø9
  BA(4Ø)=.Ø9
  BA(41)=.Ø8
  BA(42)=.Ø8
  BA(43)=.Ø8
```

```
BA(44)=.08
BA(45)=.08
BA(46)=.08
BA(47)=.08
BA(48)=.08
BA(49)=.08
BA(50)=.07
BA(51)=.07
BA(52)=.07
BA(53)=.07
BA(54)=.07
BA(55)=.07
BA(56)=.07
BA(57)=.07
BA(58)=.07
BA(59)=.07
BA(60)=.07
BA(61)=.07
BA(62)=.07
BA(63)=.07
BA(64)=.07
BA(65)=.07
BA(66)=.07
BA(67)=.07
BA(68)=.07
BA(69)=.07
BA(70)=.06
BA(71)=.06
BA(72)=.06
BA(73)=.06
BA(74)=.06
BA(75)=.06
BA(76)=.06
BA(77)=.06
BA(78)=.06
BA(79)=.06
BA(80)=.06
BA(81)=.06
BA(82)=.06
BA(83)=.06
BA(84)=.06
BA(85)=.06
BA(86)=.06
BA(87)=.06
BA(88)=.06
BA(89)=.06
BA(90)=.05
BA(91)=.05
BA(92)=.05
BA(93)=.05
BA(94)=.05
BA(95)=.05
```

```
      BA(96)=.05
      BA(97)=.05
      BA(98)=.05
      BA(99)=.05
      BA(100)=.05
      BA(101)=.05
      BA(102)=.05
      BA(103)=.05
      BA(104)=.05
      BA(105)=.05
      BA(106)=.04
      BA(107)=.04
      BA(108)=.04
      BA(109)=.04
      BA(110)=.04
      BA(111)=.04
      BA(112)=.04
      BA(113)=.04
      BA(114)=.04
      BA(115)=.04
      DO 55 JXYZ=116,250
         B(JXYZ)=0.0
         BA(JXYZ)=0.0
 55   CONTINUE
      end
C*************************************************************
      SUBROUTINE STRPOF
C     THIS SUBROUTINE IDENTIFIES LINE ITEMS OF THE
C     POPULATION WITH BOOK VALUES GREATER THEN THE
C     SAMPLING SKIP INTERVALS
      COMMON A(350),B(350),TBV(20),M,NX,TY,JJ,LL,MM,ITT
      COMMON XLTLA(11),XLTLB(11),PI(11),TYE,TE
      COMMON XMU(11),CS(11),XMCS(11),F(200,11),FA(200,11)
      COMMON YTR2CS(11),YCS(11),YTR4CS(11),YMCS(11)
      COMMON BVARY(9000),ERRARY(9000),IYI(9000),IYM(9000)
      COMMON XTR1CS(11),XTR2CS(11),XTR3CS(11),XTR4CS(11)
      COMMON TELI,TEDV,TBR,BA(350),IRND(9000),BIGBV,BIGER
      COMMON IYJ(9000),IYK(9000),IYL(9000),INDEX(9000)
      COMMON TOTARY(9000),XMDPT(20),MIDL(20),IENDPT(20)
      COMMON PR(5),XMOO(2),AVARY(9000),TNTARY(9000),ISDYPP,
     *IENDD
      COMMON BBVARY(500),AAVARY(500),EERARY(500)
      COMMON TTNTRY(500),TTOTRY(500),IINDEX(500),SKIP,
     *SIMPER,TYSKP
      COMMON/STUDY1/ ZK, SAMPER, SAMPBV, XMLE, EFF, EFFA,
     *BP, BPA
      COMMON/STUDY2/ PGW, PGWA, IM, TOTANT, ITOOT, IGY, J,
     *L, N
      COMMON/STUDY3/ IX, IY, YFL, NS, FN
      COMMON/STUDY4/ JZZ, STR, XMSTR, STRA, XMSTRA
      N=200
      JJUMP=0
```

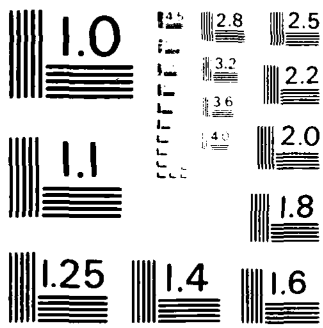
AD-A161 285 A VALIDATION OF AN ACCOUNTING UPPER ERROR LIMIT BOUND (U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF SYSTEMS AND LOGISTICS M W MELTON SEP 85 2/2

UNCLASSIFIED AFIT/GSM/LSY/85S-17 F/G 5/1 NL

END

FILMED

DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963

```
      IJUMP=0
      IBANG=0
      BIGBV=0.0
      IJACK=IENDD
      BIGER=0.0
      SIMPER=0.0
      TYSKP=TY
33    SKIP=TYSKP/N
      IBANG=0
      DO 10 I=1,IJACK
      IF (BVARY(I).GT.SKIP) THEN
      J=I
      IBANG=1
      GO TO 34
      END IF
10    CONTINUE
      IF (IBANG.EQ.0) GO TO 35
34    DO 20 K=J,IJACK
      IJUMP=IJUMP+1
      JJUMP=JJUMP+1
      BIGBV=BIGBV+BVARY(K)
      BIGER=BIGER+ERRARY(K)
      SIMPER=SIMPER+ERRARY(K)
20    CONTINUE
      TYSKP=TY-BIGBV
      N=N-JJUMP
      IJACK=IENDD-IJUMP
      JJUMP=0
      IF (IBANG.GT.0) GO TO 33
35    RETURN
      END
```

```
          do 5 ii=4,4
c         ii-four(4) main populations
          do 10 jj=1,24
c         twenty-four(24) study pops per main populations
          call input
          call cvr2r3
          call sort
          call script
 10       continue
  5       continue
          close (7)
          stop
          end
c***************************************************************
c         call to subroutines begins here.
          subroutine input
c         this subroutine puts data into arrays and computes
c         accumulators for "coverage,rel tightness two &
c         three"
          common ipop,jj,irep,ier,ied,itnt,n,te
          common im,ty,sampbv,samper,simper,zneg
          common s1(500),s2(500),s3(500),s4(500)
          common c1(500),c2(500),c3(500),c4(500)
          common ii,k,nxx
          common cvs1,cvs2,cvs3,cvs4
          common cvc1,cvc2,cvc3,cvc4
          common r1s1,r1s2,r1s3,r1s4
          common r1c1,r1c2,r1c3,r1c4
          common r2s1,r2s2,r2s3,r2s4
          common r2c1,r2c2,r2c3,r2c4
          common r3s1,r3s2,r3s3,r3s4
          common r3c1,r3c2,r3c3,r3c4
          common xmins1,xmins2,xmins3,xmins4
          common xminc1,xminc2,xminc3,xminc4
          common q1s1,q1s2,q1s3,q1s4
          common q1c1,q1c2,q1c3,q1c4
          common xmeds1,xmeds2,xmeds3,xmeds4
          common xmedc1,xmedc2,xmedc3,xmedc4
          common xmns1,xmns2,xmns3,xmns4
          common xmnc1,xmnc2,xmnc3,xmnc4
          common q3s1,q3s2,q3s3,q3s4
          common q3c1,q3c2,q3c3,q3c4
          common xmaxs1,xmaxs2,xmaxs3,xmaxs4
          common xmaxc1,xmaxc2,xmaxc3,xmaxc4
          common sds1,sds2,sds3,sds4
          common sdc1,sdc2,sdc3,sdc4
          call vzero
          do 20 k=1,500
          nxx=200
          read(ii,15) ipop,isdyp,irep,ier,ied,itnt,n,te
          read(ii,17) im,ty,sampbv,samper,simper,zneg
          read(ii,16) s1(k),s2(k),s3(k),s4(k)
```

```
      read(ii,16) c1(k),c2(k),c3(k),c4(k)
 15   format(1x,7(i4,2x),f12.2,2x)
 16   format(1x,5(f14.2,2x))
 17   format(1x,i5,2x,5(f12.2,2x))
c
c     coverage accumulators start here.
c
      if (s1(k).ge.te) cvs1=cvs1+1.0
      if (s2(k).ge.te) cvs2=cvs2+1.0
      if (s3(k).ge.te) cvs3=cvs3+1.0
      if (s4(k).ge.te) cvs4=cvs4+1.0
c
      if (c1(k).ge.te) cvc1=cvc1+1.0
      if (c2(k).ge.te) cvc2=cvc2+1.0
      if (c3(k).ge.te) cvc3=cvc3+1.0
      if (c4(k).ge.te) cvc4=cvc4+1.0
c
c     rel tightness two(2) accumulators start here!
c
      r2s1=r2s1+(s1(k)/s1(k))
      r2s2=r2s2+(s1(k)/s2(k))
      r2s3=r2s3+(s3(k)/s3(k))
      r2s4=r2s4+(s3(k)/s4(k))
c
      r2c1=r2c1+(s1(k)/c1(k))
      r2c2=r2c2+(s2(k)/c2(k))
      r2c3=r2c3+(s3(k)/c3(k))
      r2c4=r2c4+(s4(k)/c4(k))
c
 20   continue
      return
      end
c*********************************************************
      subroutine cvr2r3
c     subroutine calculates coverage,rel tightness
c     two(2)
      common ipop,jj,irep,ier,ied,itnt,n,te
      common im,ty,sampbv,samper,simper,zneg
      common s1(500),s2(500),s3(500),s4(500)
      common c1(500),c2(500),c3(500),c4(500)
      common ii,k,nxx
      common cvs1,cvs2,cvs3,cvs4
      common cvc1,cvc2,cvc3,cvc4
      common r1s1,r1s2,r1s3,r1s4
      common r1c1,r1c2,r1c3,r1c4
      common r2s1,r2s2,r2s3,r2s4
      common r2c1,r2c2,r2c3,r2c4
      common r3s1,r3s2,r3s3,r3s4
      common r3c1,r3c2,r3c3,r3c4
      common xmins1,xmins2,xmins3,xmins4
      common xminc1,xminc2,xminc3,xminc4
      common q1s1,q1s2,q1s3,q1s4
```

```
      common qlcl,qlc2,qlc3,qlc4
      common xmedsl,xmeds2,xmeds3,xmeds4
      common xmedcl,xmedc2,xmedc3,xmedc4
      common xmnsl,xmns2,xmns3,xmns4
      common xmncl,xmnc2,xmnc3,xmnc4
      common q3sl,q3s2,q3s3,q3s4
      common q3cl,q3c2,q3c3,q3c4
      common xmaxsl,xmaxs2,xmaxs3,xmaxs4
      common xmaxcl,xmaxc2,xmaxc3,xmaxc4
      common sdsl,sds2,sds3,sds4
      common sdcl,sdc2,sdc3,sdc4
      x=500.0
c     coverage calculations begin here!
      cvsl=cvsl/x
      cvs2=cvs2/x
      cvs3=cvs3/x
      cvs4=cvs4/x
c
      cvcl=cvcl/x
      cvc2=cvc2/x
      cvc3=cvc3/x
      cvc4=cvc4/x
c
c     rel tightness two(2) calculations begin here!
      r2sl=r2sl/x
      r2s2=r2s2/x
      r2s3=r2s3/x
      r2s4=r2s4/x
c
      r2cl=r2cl/x
      r2c2=r2c2/x
      r2c3=r2c3/x
      r2c4=r2c4/x
c
      return
      end
c***********************************************************
      subroutine sort
c     subroutine sorts all bound arrays from lowest to
c        highest value and calls subroutine statrl!
      common ipop,jj,irep,ier,ied,itnt,n,te
      common im,ty,sampbv,samper,simper,zneg
      common sl(500),s2(500),s3(500),s4(500)
      common cl(500),c2(500),c3(500),c4(500)
      common ii,k,nxx
      common cvsl,cvs2,cvs3,cvs4
      common cvcl,cvc2,cvc3,cvc4
      common rlsl,rls2,rls3,rls4
      common rlcl,rlc2,rlc3,rlc4
      common r2sl,r2s2,r2s3,r2s4
      common r2cl,r2c2,r2c3,r2c4
      common r3sl,r3s2,r3s3,r3s4
```

```
      common r3c1,r3c2,r3c3,r3c4
      common xmins1,xmins2,xmins3,xmins4
      common xminc1,xminc2,xminc3,xminc4
      common q1s1,q1s2,q1s3,q1s4
      common q1c1,q1c2,q1c3,q1c4
      common xmeds1,xmeds2,xmeds3,xmeds4
      common xmedc1,xmedc2,xmedc3,xmedc4
      common xmns1,xmns2,xmns3,xmns4
      common xmnc1,xmnc2,xmnc3,xmnc4
      common q3s1,q3s2,q3s3,q3s4
      common q3c1,q3c2,q3c3,q3c4
      common xmaxs1,xmaxs2,xmaxs3,xmaxs4
      common xmaxc1,xmaxc2,xmaxc3,xmaxc4
      common sds1,sds2,sds3,sds4
      common sdc1,sdc2,sdc3,sdc4
      nn=500
      nm1=nn-1
      do 1 j=1,nm1
      nmj=nn-j
      do 2 i=1,nmj
      if (s1(i).gt.s1(i+1)) then
      tmps1=s1(i)
      s1(i)=s1(i+1)
      s1(i+1)=tmps1
      end if
c
      if (s2(i).gt.s2(i+1)) then
      tmps2=s2(i)
      s2(i)=s2(i+1)
      s2(i+1)=tmps2
      end if
c
      if (s3(i).gt.s3(i+1)) then
      tmps3=s3(i)
      s3(i)=s3(i+1)
      s3(i+1)=tmps3
      end if
c
      if (s4(i).gt.s4(i+1)) then
      tmps4=s4(i)
      s4(i)=s4(i+1)
      s4(i+1)=tmps4
      end if
c
      if (c1(i).gt.c1(i+1)) then
      tmpc1=c1(i)
      c1(i)=c1(i+1)
      c1(i+1)=tmpc1
      end if
c
      if (c2(i).gt.c2(i+1)) then
      tmpc2=c2(i)
```

```
      c2(i)=c2(i+1)
      c2(i+1)=tmpc2
      end if
c
      if (c3(i).gt.c3(i+1)) then
      tmpc3=c3(i)
      c3(i)=c3(i+1)
      c3(i+1)=tmpc3
      end if
c
      if (c4(i).gt.c4(i+1)) then
      tmpc4=c4(i)
      c4(i)=c4(i+1)
      c4(i+1)=tmpc4
      end if
c
 2    continue
 1    continue
      call statrl
      return
      end
c************************************************************
      subroutine statrl
c     subroutine computes statistics for each bound and
c     calculates rel tightness one(1)
      common ipop,jj,irep,ier,ied,itnt,n,te
      common im,ty,sampbv,samper,simper,zneg
      common s1(500),s2(500),s3(500),s4(500)
      common c1(500),c2(500),c3(500),c4(500)
      common ii,k,nxx
      common cvs1,cvs2,cvs3,cvs4
      common cvc1,cvc2,cvc3,cvc4
      common r1s1,r1s2,r1s3,r1s4
      common r1c1,r1c2,r1c3,r1c4
      common r2s1,r2s2,r2s3,r2s4
      common r2c1,r2c2,r2c3,r2c4
      common r3s1,r3s2,r3s3,r3s4
      common r3c1,r3c2,r3c3,r3c4
      common xmins1,xmins2,xmins3,xmins4
      common xminc1,xminc2,xminc3,xminc4
      common q1s1,q1s2,q1s3,q1s4
      common q1c1,q1c2,q1c3,q1c4
      common xmeds1,xmeds2,xmeds3,xmeds4
      common xmedc1,xmedc2,xmedc3,xmedc4
      common xmns1,xmns2,xmns3,xmns4
      common xmnc1,xmnc2,xmnc3,xmnc4
      common q3s1,q3s2,q3s3,q3s4
      common q3c1,q3c2,q3c3,q3c4
      common xmaxs1,xmaxs2,xmaxs3,xmaxs4
      common xmaxc1,xmaxc2,xmaxc3,xmaxc4
      common sds1,sds2,sds3,sds4
      common sdc1,sdc2,sdc3,sdc4
```

```
      nn=500
c
      xmeds1=(s1(nn/2)+s1((nn/2)+1))/2.0
      xmeds2=(s2(nn/2)+s2((nn/2)+1))/2.0
      xmeds3=(s3(nn/2)+s3((nn/2)+1))/2.0
      xmeds4=(s4(nn/2)+s4((nn/2)+1))/2.0
c
      xmedc1=(c1(nn/2)+c1((nn/2)+1))/2.0
      xmedc2=(c2(nn/2)+c2((nn/2)+1))/2.0
      xmedc3=(c3(nn/2)+c3((nn/2)+1))/2.0
      xmedc4=(c4(nn/2)+c4((nn/2)+1))/2.0
c
      xmins1=s1(1)
      xmins2=s2(1)
      xmins3=s3(1)
      xmins4=s4(1)
c
      xminc1=c1(1)
      xminc2=c2(1)
      xminc3=c3(1)
      xminc4=c4(1)
c
      xmaxs1=s1(nn)
      xmaxs2=s2(nn)
      xmaxs3=s3(nn)
      xmaxs4=s4(nn)
c
      xmaxc1=c1(nn)
      xmaxc2=c2(nn)
      xmaxc3=c3(nn)
      xmaxc4=c4(nn)
c
      q1s1=s1(nn/4)
      q1s2=s2(nn/4)
      q1s3=s3(nn/4)
      q1s4=s4(nn/4)
c
      q1c1=c1(nn/4)
      q1c2=c2(nn/4)
      q1c3=c3(nn/4)
      q1c4=c4(nn/4)
c
      q3s1=s1(nn*3/4)
      q3s2=s2(nn*3/4)
      q3s3=s3(nn*3/4)
      q3s4=s4(nn*3/4)
c
      q3c1=c1(nn*3/4)
      q3c2=c2(nn*3/4)
      q3c3=c3(nn*3/4)
      q3c4=c4(nn*3/4)
c
```

```
      do 3 i=1,500
c
      xmns1=xmns1+s1(i)
      xmns2=xmns2+s2(i)
      xmns3=xmns3+s3(i)
      xmns4=xmns4+s4(i)
c
      xmnc1=xmnc1+c1(i)
      xmnc2=xmnc2+c2(i)
      xmnc3=xmnc3+c3(i)
      xmnc4=xmnc4+c4(i)
c
 3    continue
      xi=500.0
c
      xmns1=xmns1/xi
      xmns2=xmns2/xi
      xmns3=xmns3/xi
      xmns4=xmns4/xi
c
      xmnc1=xmnc1/xi
      xmnc2=xmnc2/xi
      xmnc3=xmnc3/xi
      xmnc4=xmnc4/xi
c
      do 4 i=1,500
c
      sigs1=sigs1+((s1(i)-xmns1))**2
      sigs2=sigs2+((s2(i)-xmns2))**2
      sigs3=sigs3+((s3(i)-xmns3))**2
      sigs4=sigs4+((s4(i)-xmns4))**2
c
      sigc1=sigc1+((c1(i)-xmnc1))**2
      sigc2=sigc2+((c2(i)-xmnc2))**2
      sigc3=sigc3+((c3(i)-xmnc3))**2
      sigc4=sigc4+((c4(i)-xmnc4))**2
c
 4    continue
c
      sds1=(sigs1/(xi-1))**0.5
      sds2=(sigs2/(xi-1))**0.5
      sds3=(sigs3/(xi-1))**0.5
      sds4=(sigs4/(xi-1))**0.5
c
      sdc1=(sigc1/(xi-1))**0.5
      sdc2=(sigc2/(xi-1))**0.5
      sdc3=(sigc3/(xi-1))**0.5
      sdc4=(sigc4/(xi-1))**0.5
c
      rls1=xmeds1/xmeds1
      rls2=xmeds1/xmeds2
      rls3=xmeds3/xmeds3
```

```
      rls4=xmeds3/xmeds4
c
      rlc1=xmeds1/xmedc1
      rlc2=xmeds1/xmedc2
      rlc3=xmeds1/xmedc3
      rlc4=xmeds1/xmedc4
c
      return
      end
c*******************************************************
      subroutine script
c     subroutine writes output to file called
c     analysis1.f
      common ipop,jj,irep,ier,ied,itnt,n,te
      common im,ty,sampbv,samper,simper,zneg
      common s1(500),s2(500),s3(500),s4(500)
      common c1(500),c2(500),c3(500),c4(500)
      common ii,k,nxx
      common cvs1,cvs2,cvs3,cvs4
      common cvc1,cvc2,cvc3,cvc4
      common rls1,rls2,rls3,rls4
      common rlc1,rlc2,rlc3,rlc4
      common r2s1,r2s2,r2s3,r2s4
      common r2c1,r2c2,r2c3,r2c4
      common r3s1,r3s2,r3s3,r3s4
      common r3c1,r3c2,r3c3,r3c4
      common xmins1,xmins2,xmins3,xmins4
      common xminc1,xminc2,xminc3,xminc4
      common q1s1,q1s2,q1s3,q1s4
      common q1c1,q1c2,q1c3,q1c4
      common xmeds1,xmeds2,xmeds3,xmeds4
      common xmedc1,xmedc2,xmedc3,xmedc4
      common xmns1,xmns2,xmns3,xmns4
      common xmnc1,xmnc2,xmnc3,xmnc4
      common q3s1,q3s2,q3s3,q3s4
      common q3c1,q3c2,q3c3,q3c4
      common xmaxs1,xmaxs2,xmaxs3,xmaxs4
      common xmaxc1,xmaxc2,xmaxc3,xmaxc4
      common sds1,sds2,sds3,sds4
      common sdc1,sdc2,sdc3,sdc4
c
      write(7,*)
      write(7,*)
      write(7,301) ipop,jj,ier,ied,itnt,nxx,te
      write(7,*)
      write(7,302) cvs1,cvs2,cvs3,cvs4
      write(7,302) cvc1,cvc2,cvc3,cvc4
      write(7,*)
      write(7,302) rls1,rls2,rls3,rls4
      write(7,302) rlc1,rlc2,rlc3,rlc4
      write(7,*)
      write(7,302) r2s1,r2s2,r2s3,r2s4
```

```
      write(7,302) r2cl,r2c2,r2c3,r2c4
      write(7,*)
      write(7,303) xminsl,xmins2,xmins3,xmins4
      write(7,303) xmincl,xminc2,xminc3,xminc4
      write(7,*)
      write(7,303) qlsl,qls2,qls3,qls4
      write(7,303) qlcl,qlc2,qlc3,qlc4
      write(7,*)
      write(7,303) xmnsl,xmns2,xmns3,xmns4
      write(7,303) xmncl,xmnc2,xmnc3,xmnc4
      write(7,*)
      write(7,303) xmedsl,xmeds2,xmeds3,xmeds4
      write(7,303) xmedcl,xmedc2,xmedc3,xmedc4
      write(7,*)
      write(7,303) q3sl,q3s2,q3s3,q3s4
      write(7,303) q3cl,q3c2,q3c3,q3c4
      write(7,*)
      write(7,303) xmaxsl,xmaxs2,xmaxs3,xmaxs4
      write(7,303) xmaxcl,xmaxc2,xmaxc3,xmaxc4
      write(7,*)
      write(7,303) sdsl,sds2,sds3,sds4
      write(7,303) sdcl,sdc2,sdc3,sdc4
 301  format(1x,6(i4,2x),f12.2,2x)
 302  format(1x,5(f12.4,2x))
 303  format(1x,5(f12.2,2x))
      return
      end
c*************************************************************
      subroutine vzero
c     subroutine initializes and zeros out all variables
c
      common ipop,jj,irep,ier,ied,itnt,n,te
      common im,ty,sampbv,samper,simper,zneg
      common sl(500),s2(500),s3(500),s4(500)
      common cl(500),c2(500),c3(500),c4(500)
      common ii,k,nxx
      common cvsl,cvs2,cvs3,cvs4
      common cvcl,cvc2,cvc3,cvc4
      common rlsl,rls2,rls3,rls4
      common rlcl,rlc2,rlc3,rlc4
      common r2sl,r2s2,r2s3,r2s4
      common r2cl,r2c2,r2c3,r2c4
      common r3sl,r3s2,r3s3,r3s4
      common r3cl,r3c2,r3c3,r3c4
      common xminsl,xmins2,xmins3,xmins4
      common xmincl,xminc2,xminc3,xminc4
      common qlsl,qls2,qls3,qls4
      common qlcl,qlc2,qlc3,qlc4
      common xmedsl,xmeds2,xmeds3,xmeds4
      common xmedcl,xmedc2,xmedc3,xmedc4
      common xmnsl,xmns2,xmns3,xmns4
      common xmncl,xmnc2,xmnc3,xmnc4
```

```
            common q3s1,q3s2,q3s3,q3s4
            common q3c1,q3c2,q3c3,q3c4
            common xmaxs1,xmaxs2,xmaxs3,xmaxs4
            common xmaxc1,xmaxc2,xmaxc3,xmaxc4
            common sds1,sds2,sds3,sds4
            common sdc1,sdc2,sdc3,sdc4
c
            ipop=0
            irep=0
            ier=0
            ied=0
            itnt=0
            n=0
            te=0.0
c
            do 1 i=1,500
            s1(i)=0.0
            s2(i)=0.0
            s3(i)=0.0
            s4(i)=0.0
c
            c1(i)=0.0
            c2(i)=0.0
            c3(i)=0.0
            c4(i)=0.0
c
 1          continue
c
            cvs1=0.0
            cvs2=0.0
            cvs3=0.0
            cvs4=0.0
c
            cvc1=0.0
            cvc2=0.0
            cvc3=0.0
            cvc4=0.0
c
            r1s1=0.0
            r1s2=0.0
            r1s3=0.0
            r1s4=0.0
c
            r1c1=0.0
            r1c2=0.0
            r1c3=0.0
            r1c4=0.0
c
            r2s1=0.0
            r2s2=0.0
            r2s3=0.0
            r2s4=0.0
```

```
c
      r2c1=0.0
      r2c2=0.0
      r2c3=0.0
      r2c4=0.0
c
      xmins1=0.0
      xmins2=0.0
      xmins3=0.0
      xmins4=0.0
c
      xminc1=0.0
      xminc2=0.0
      xminc3=0.0
      xminc4=0.0
c
      q1s1=0.0
      q1s2=0.0
      q1s3=0.0
      q1s4=0.0
c
      q1c1=0.0
      q1c2=0.0
      q1c3=0.0
      q1c4=0.0
c
      xmeds1=0.0
      xmeds2=0.0
      xmeds3=0.0
      xmeds4=0.0
c
      xmedc1=0.0
      xmedc2=0.0
      xmedc3=0.0
      xmedc4=0.0
c
      xmns1=0.0
      xmns2=0.0
      xmns3=0.0
      xmns4=0.0
c
      xmnc1=0.0
      xmnc2=0.0
      xmnc3=0.0
      xmnc4=0.0
c
      q3s1=0.0
      q3s2=0.0
      q3s3=0.0
      q3s4=0.0
c
      q3c1=0.0
```

```
      q3c2=0.0
      q3c3=0.0
      q3c4=0.0
c
      xmaxs1=0.0
      xmaxs2=0.0
      xmaxs3=0.0
      xmaxs4=0.0
c
      xmaxc1=0.0
      xmaxc2=0.0
      xmaxc3=0.0
      xmaxc4=0.0
c
      sigs1=0.0
      sigs2=0.0
      sigs3=0.0
      sigs4=0.0
c
      sigc1=0.0
      sigc2=0.0
      sigc3=0.0
      sigc4=0.0
c
      sds1=0.0
      sds2=0.0
      sds3=0.0
      sds4=0.0
c
      sdc1=0.0
      sdc2=0.0
      sdc3=0.0
      sdc4=0.0
c
      return
      end
```

Bibliography

1. Garstka, S. J. "Models for Computing Upper Error Limits in Dollar-Unit Sampling," Journal of Accounting Research, 15: 179-192 (Autumn 1977).

2. Johnson, J. R., R. A. Leitch, and John Neter "Characteristics of Errors in Accounts Receivable and Inventory Audits," The Accounting Review, 56: 270-293 (April 1981).

3. Neter, John, R. A. Leitch, and S. E. Fienburg "Dollar Unit Sampling: Multinomial Bounds for Total Overstatements and Understatements Errors," The Accounting Review, 53: 77-93 (January 1978).

4. Reneau, J. H. "CAV Bounds in Dollar Unit Sampling: Some Simulation Results," The Accounting Review, 53: 669-680 (July 1978).

5. Godfrey, James T. and Richard W. Andrews "A Finite Bayesian Model for Compliance Testing," Journal of Accounting Research, 20: 304-315 (Autumn 1982).

6. Leslie, Donald A., Albert D. Teitlebaum and Rodney J. Anderson. Dollar Unit Sampling: A Practical Guide for Auditors. Toronto: Copp Clark Pittman Publishers Inc., 1979.

7. Godfrey, James T. and John Neter "Bayesian Bounds for Monetary-Unit-Sampling in Accounting and Auditing," Audit Research Working Paper Series Report Number 83-001, University of Georgia Audit Research Center, 1983.

8. Cox, D. R. and E. J. Snell "On Sampling and the Estimation of Rare Errors," Biometrika, 66: 125-132 (1979).

9. Markland, Robert E. Topics in Management Science (Second Edition). New York: John Wiley & Sons Inc., 1983.

10. McCray, John H. "A Quasi-Bayesian Audit Risk Model for Dollar Unit Sampling," The Accounting Review, 59: 35-51 (January 1984).

11. Neter, J. and J. K. Loebbeccki. Behavior of Major Statistical Estimators in Sampling Accounting Populations. New York: AICPA Inc., 1975.

VITA

First Lieutenant Michael W. Helton was born on 13 August 1959 in Springfield, Massachusetts. He Graduated from Cathedral High School in 1977 and attended Holyoke Community College for one year. From there he went to the United States Air Force Academy from which he received the degree of Bachelor of Science in Management in June 1982. Upon graduating, he was assigned to the Cost and Management Analysis branch at Randolph AFB, Texas, until entering the School of Systems and Logistics, Air Force Institute of Technology, in May 1984.

Permanent address: 2621 Pebble Valley
San Antonio, Texas 78232

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

AD-A161 205

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION: UNCLASSIFIED

1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION/DOWNGRADING SCHEDULE

3. DISTRIBUTION/AVAILABILITY OF REPORT: Approved for public release; distribution unlimited.

4. PERFORMING ORGANIZATION REPORT NUMBER(S): AFIT/GSM/LSY/85S-17

5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION: School of Systems and Logistics

6b. OFFICE SYMBOL (If applicable): AFIT/LSQ

7a. NAME OF MONITORING ORGANIZATION

6c. ADDRESS (City, State and ZIP Code): Air Force Institute of Technology, Wright-Patterson AFB, Ohio 45433

7b. ADDRESS (City, State and ZIP Code)

8a. NAME OF FUNDING/SPONSORING ORGANIZATION

8b. OFFICE SYMBOL (If applicable)

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c. ADDRESS (City, State and ZIP Code)

10. SOURCE OF FUNDING NOS.

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO.

11. TITLE (Include Security Classification): See Box 19

12. PERSONAL AUTHOR(S): Michael W. Helton, B.S., 1st Lt, USAF

13a. TYPE OF REPORT: MS Thesis

13b. TIME COVERED: FROM ______ TO ______

14. DATE OF REPORT (Yr., Mo., Day): 1985 September

15. PAGE COUNT: 114

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB. GR.
05	01	
14	01	

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number): Accounting, Auditing, Sampling, Model Theory

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Title: A VALIDATION OF AN ACCOUNTING UPPER ERROR LIMIT BOUND

Thesis Chairman: Jeffrey J. Phillips, Lt Col, USAF

Approved for public release: IAW AFR 190-1.
11 SEPT 85
LYNN E. WOLAVER
Dean for Research and Professional Development
Air Force Institute of Technology (ATC)
Wright-Patterson AFB OH 45433

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT: UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS ☐

21. ABSTRACT SECURITY CLASSIFICATION: UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL: Jeffrey J. Phillips, Lt Col, USAF

22b. TELEPHONE NUMBER (Include Area Code): 513-255-4845

22c. OFFICE SYMBOL: AFIT/LSY

DD FORM 1473, 83 APR   EDITION OF 1 JAN 73 IS OBSOLETE.

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

The purpose of this research was to examine the validity of an accounting upper error limit bound. The bound examined was the DUS-cell method suggested by Leslie, Teitlebaum, and Anderson which was supposed to reduce bound conservativism and produce actual confidence levels closer to the nominal confidence levels.

The analysis was accomplished by examining the robustness, the relative tightness, and the effects of error rate, error clustering, mean taint, and error amount intensity on the coverage and relative tightness of the DUS-cell bound. The results of this research indicate that the DUS-cell bound is not robust and is tighter than the Stringer bound. The results also demonstrated that error rate has the greatest effect on coverage and relative tightness, error clustering has some effect, and mean taint has little effect. The results also indicate that error amount intensity, a population characteristic, affects coverage and relative tightness of the DUS-cell bound significantly.

END

FILMED

12-85

DTIC